NASA Conference Publication 2459

Computer Sciences and Data Systems

Volume 1

(NASA-CP-2459-Vol-1) COMPUTER SCIENCES AND DATA SYSTEMS, VOLUME 1 (National Aeronautics and Space Administration) 356 p CSCL 09B
N87-19931
Unclas
H1/61 43656

Proceedings of a symposium held at
the National Conference Center in
Williamsburg, Virginia
November 18-20, 1986

NASA

NASA Conference Publication 2459

Computer Sciences and Data Systems

Volume 1

Proceedings of a symposium held at
the National Conference Center in
Williamsburg, Virginia
November 18–20, 1986

NASA
**National Aeronautics
and Space Administration**

**Scientific and Technical
Information Branch**

1987

OAST Computer Science/Data Systems Technical Symposium

Table of Contents

Volume 1: Computer Science

Title of Presentation — Page

Table of Contents iii
Introduction v

Section 1: Software Engineering

Software Management Environment for NASA 1-1
SAGA: An Integrated Software Development Environment 1-16
Cognitive-Based Analysis of Software Documentation 1-28
Software Life Cycle Simulation (SLICS) Modeling 1-49
Software Engineering with Reusable Components Task 1-59
Learning the Behavior of Software Systems From Executable Specifications 1-66

Section 2: University Grants

JSC Sponsored Research Activity at University of Houston at Clear Lake 1-85
Illinois Computing Laboratory of Aerospace Systems and Software (ICLASS) 1-109
Overview of ICLASS Research: Reliable and Parallel Computing 1-123
Center for Aeronautics and Space Information Sciences (CASIS) 1-136

Section 3: Institutes

Institute for Computer Applications in Science and Engineering 3-162
Research Institute for Advanced Computer Science 1-173
NASA Center of Excellence in Space Data Information Sciences (CESDIS) 1-197

PRECEDING PAGE BLANK NOT FILMED

Section 4: Applications

Concurrent Processing Algorithm Design (CPAD)..................... 1-220
Performance Critical Decisions in Parallel Scientific Computations... 1-231
Sparse Distributed Memory.. 1-241
Distributed Operating Systems - Fact or Fantasy?.................. 1-261
Distributed Access View Integrated Database (DAVID) System........ 1-269
Intelligent Data Management Processes............................. 1-278
An Expert System For the Analysis of Imaging Spectrometer Data.... 1-298
Fault-Tolerant Software: Modeling and Validation................. 1-312
The Implementation and Use of ADA for Fault-Tolerant Distributed Systems... 1-325
Architecture Research at ARC.. 1-332

Volume 2: Data Systems

Optical Archival Data Storage System.............................. 2-1
Information Network Architectures................................... 2-16
NASA OAST VHSIC Technology.. 2-31
Semiconductor Laser and Fiber Optics Technology................... 2-52
High Speed Taken Ring Performance Analysis........................ 2-63
STAR* Bus.. 2-84
A Distributed Processing Network Simulator (DPNS)................. 2-107
Distributed Processing Concepts..................................... 2-131
A Systems Level Approach to Distributed Processing................ 2-143
Spaceborne Optical Disk Controller Development.................... 2-173
On-Board GaAs Processor Development................................ 2-186
MAX: A High-Speed, General-Purpose Multicomputer for Space Applications.. 2-208
Massively Parallel Processor (MPP)................................. 2-234
Advanced Digital SAR Processor (ADSP).............................. 2-248
Flight SAR Processor Study... 2-264
Electronic Associative Memory Based on Hopfield's Neural Network Model.. 2-277
NASA Computer Science Research Program Plan Update................ 2-317
Closing Remarks.. 2-324
Attendance List.. 2-325

INTRODUCTION

The Computer Sciences and Data Systems Technical Symposium was held to respond to the communications challenges posed by the rapidly advancing technical arena surrounding NASA personnel. This was the third meeting in what will be periodic gatherings and was hosted by LaRC. Jerry Creedon, Director for Flight Systems at LaRC, performed the welcoming ceremony, and opening remarks were made by Lee Holcomb, Director of Information Sciences and Human Factors at NASA Headquarters.

The intended purpose of these symposia is to bring NASA people together to present their progress, to air their thinking and, in general, to discuss the nature and results of their work within the agency on a wholly technical level. These meetings are not intended as a forum for program reviews, budget presentations or advocacy hearings. NASA personnel have long been recognized as prolific contributors to the journals of technical societies and organizations within the aerospace community. Meetings such as this, organized to improve the interchange of technical information and understanding within NASA, have resulted in valuable connections. These meetings will be continued to be held at approximately 18 month intervals. The Proceedings of the November 1986 Computer Sciences and Data Systems Technical Symposium are presented to provide continuity from one meeting to the next, and to serve as a technical blueprint regarding expected content.

NASA

MISSION OPERATIONS
AND DATA SYSTEMS DIRECTORATE

GSFC

SOFTWARE MANAGEMENT ENVIRONMENT FOR NASA

JOHN DALTON

DRO059D A-1

MO&DS
DIRECTORATE
CODE 500

SOFTWARE MANAGEMENT ENVIRONMENT
FOR NASA

SOFTWARE MANAGEMENT ENVIRONMENT FOR NASA

- OBJECTIVE
  - DEVELOP, ASSESS AND IMPLEMENT SOFTWARE MANAGEMENT AIDS (TOOLS, MEASURES, TECHNIQUES)
  - LEADING TO AN ENVIRONMENT PRODUCING SOFTWARE OF "INCREASED QUALITY" (RELIABILITY AND LIFE CYCLE COST)
- AREAS OF INVESTIGATION
  - DESIGN AND SPECIFICATION MEASURES
  - MANAGEMENT TOOLS (INCLUDING RAPID PROTOTYPING AIDS)

AREAS OF CONSIDERATION

- DESIGN/SPECIFICATION MEASURES
  - CAN WE DETERMINE "QUALITY" OF DESIGN (OR SPECS)?
  - WHAT IS "QUALITY" FOR DESIGN?
  - HOW DO WE DETERMINE TRADE-OFFS FOR VARIOUS DESIGN APPROACHES?
  - CAN I DETERMINE EARLY WHAT PART OF THE SYSTEM IS "EASY" OR "HARD"?
- MANAGEMENT TOOLS
  - AUTOMATICALLY DETERMINE QUALITY OF DESIGN
  - AUTOMATICALLY DETERMINE "IMPROVED" DESIGN
  - EVALUATE SPECS
  - GIVEN EXISTING DEVELOPMENT INFORMATION, PREDICT AND ASSESS
- RAPID PROTOTYPING TECHNOLOGY
  - IS RAPID PROTOTYPING EFFECTIVE? IF SO, WHEN, AND WITH WHAT TOOLS?
  - CAN WE LOOK AT ESSENTIAL SYSTEM CHARACTERISTICS EARLY AND EASILY?

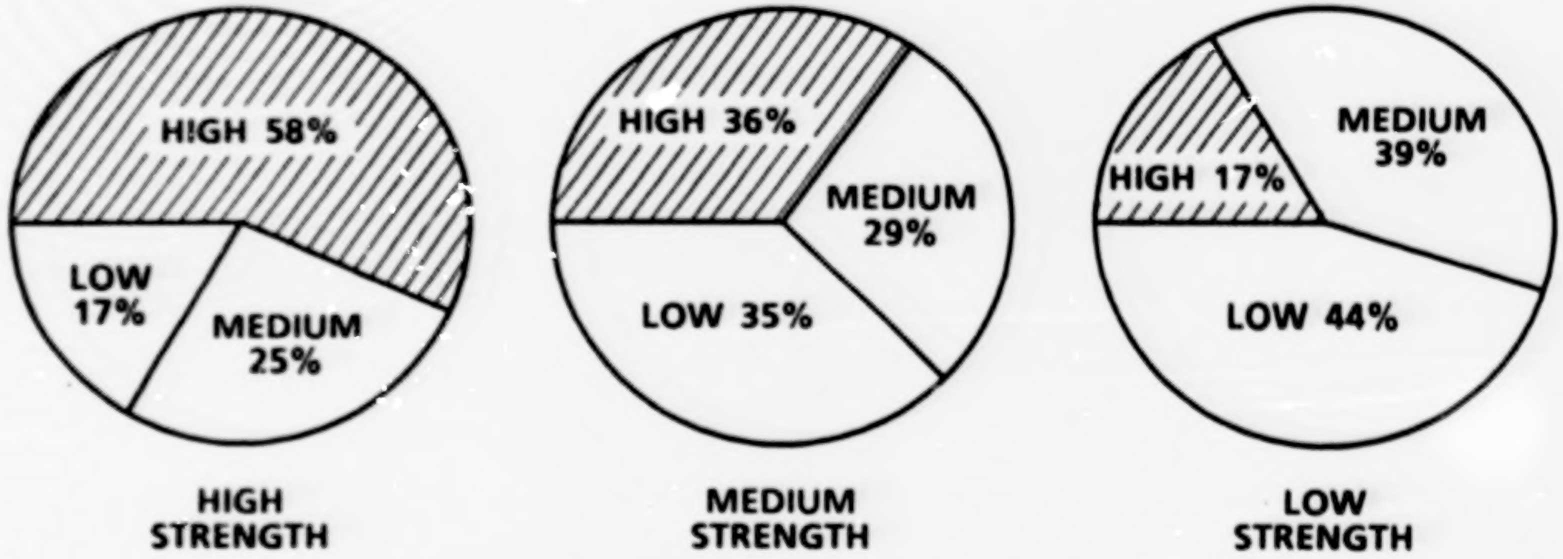
DR0059D A-2

MO&DS DIRECTORATE
CODE 500

SOFTWARE MANAGEMENT ENVIRONMENT FOR NASA

GSFC

STRENGTH AS A DESIGN MEASURE



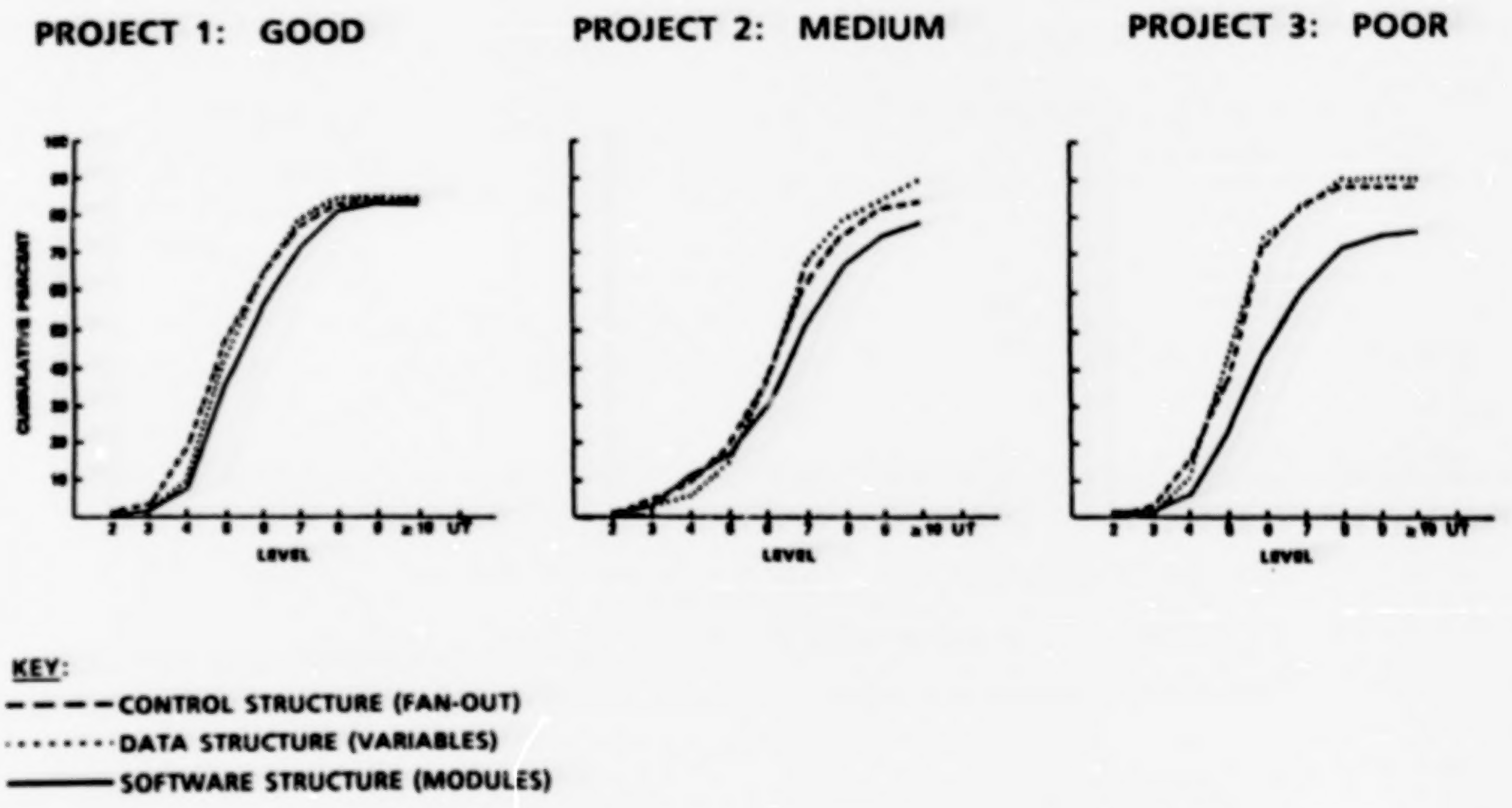
HIGH STRENGTH IMPLIES HIGH RELIABILITY

BASED ON:
- * 480 MODULES
- * 3 PROJECTS

RELIABILITY:
HIGH ERROR = 0/1000 L.O.C.
MED ERROR.LE.2/1000 L.O.C.
LOW ERROR.GT.2/1000 L.O.C.

DRO059D A-2

DESIGN IS A PARTITIONING OF STRUCTURE



DRO059D A-2

S/W SPECIFICATION MEASURES: PROPOSE A NEW REPRESENTATION

COMPOSITE SPECIFICATION MODEL (CSM)

RATIONALE: REQUIREMENTS FOR COMPLEX SOFTWARE NEED TO BE SPECIFIED FROM MULTIPLE VIEWPOINTS

VIEWPOINT	NOTATION
• FUNCTIONAL	• DATA FLOW
• CONTEXTUAL	• ENTITY/RELATIONSHIP
• DYNAMIC	• STATE/TRANSITION

STUDY OF TWO FLIGHT DYNAMICS PROJECTS: CONCLUSIONS

- OBJECTIVE SPECIFICATION MEASURES NEED DISCIPLINED REPRESENTATION OF REQUIREMENTS
- BUILDING THE CSM IS FEASIBLE
  - YIELDS OBJECTIVE SPECIFICATION MEASURES
  - MULTIPLE VIEWS ARE MORE REVEALING
  - MORE EFFECTIVE REPRESENTATION TO BEGIN DESIGN
- CAPTURING THE CONTEXT OF A SYSTEM IS BENEFICIAL
  - SOURCE OF CHANGES TO THE SYSTEM
  - LOGICAL PREDECESSOR OF OBJECT-ORIENTED DESIGN

DRO0590 A-2

DYNAMIC MANAGEMENT INFORMATION TOOL: THE IDEA

INPUT	OUTPUT
1. VERIFIED MEASURE/MODELS FOR DEVELOPMENT (E.G., 40-20-40 RULE OR RAYLEIGH CURVE)	1. PREDICT (E.G., WHEN WILL PROJECT BE COMPLETE?)
2. PAST PROJECT HISTORIES (E.G., STAFFING PROFILES)	2. ASSESS (E.G., TESTING PROCEDURES ARE BAD)
3. VERIFIED "RULES" OF SOFTWARE DEVELOPMENT (E.G., IF EXCESSIVE ECR'S THEN SPECS ARE OF POOR QUALITY)	3. COMPARE (E.G., RELATIVE TO PAST PROJECTS, THE CODE DEVELOPMENT RATE IS VERY LOW)
4. CURRENT PROJECT DEVELOPMENT DATA (E.G., STAFFING, CHANGES, RESOURCE CONSUMPTION)	4. SELECT/CONTROL (E.G., USE TIGHTER TESTING STANDARDS FOR THIS PROJECT)

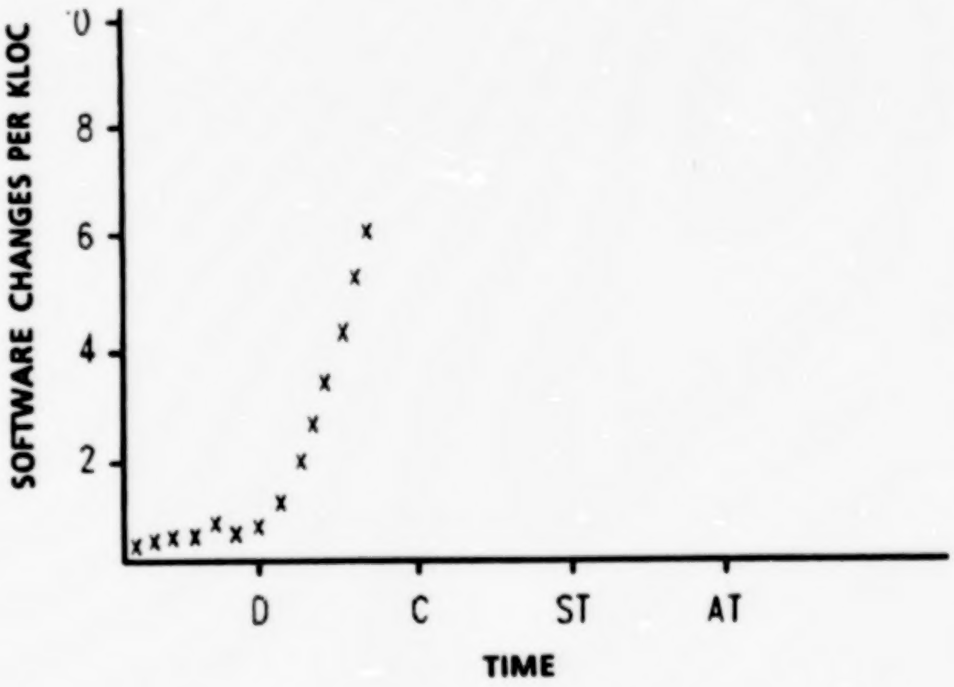
MO&DS
DIRECTORATE
CODE 500

SOFTWARE MANAGEMENT ENVIRONMENT
FOR NASA

GSFC

DYNAMITE SCENARIO UTILIZING EXPERT SYSTEM

STEP 1
RETRIEVE DATA
FROM DYNAMIC
PROJECT FILE



DRO059D A-2

MO&DS DIRECTORATE

CODE 500

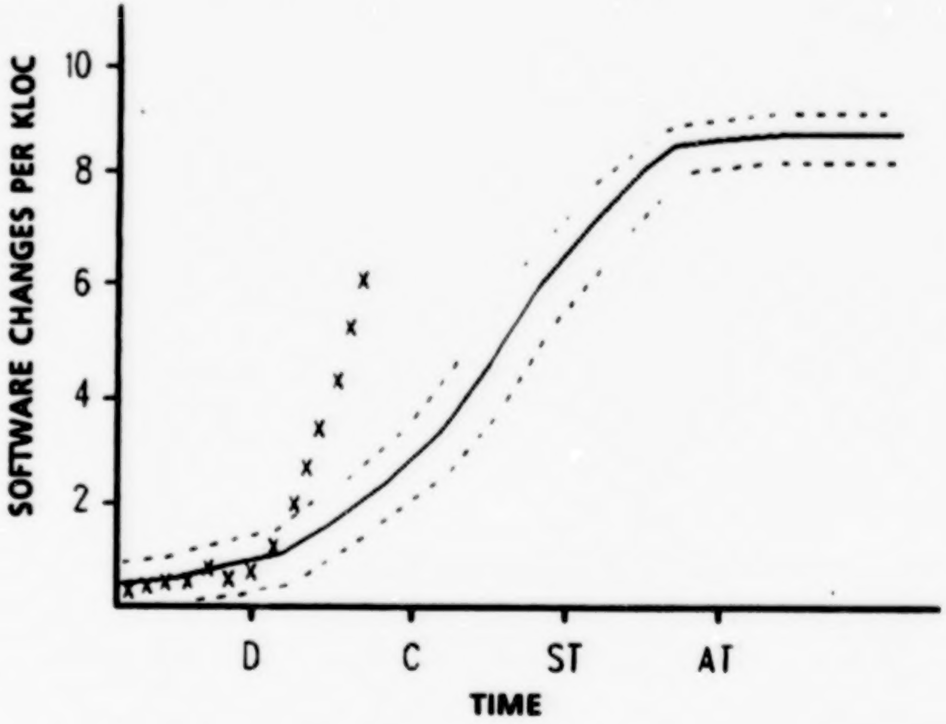
SOFTWARE MANAGEMENT ENVIRONMENT
FOR NASA

GSFC

DYNAMITE SCENARIO UTILIZING EXPERT SYSTEM

STEP 1
RETRIEVE DATA FROM DYNAMIC PROJECT FILE

STEP 2
RETRIEVE MODE OF SEL EXPERIENCE



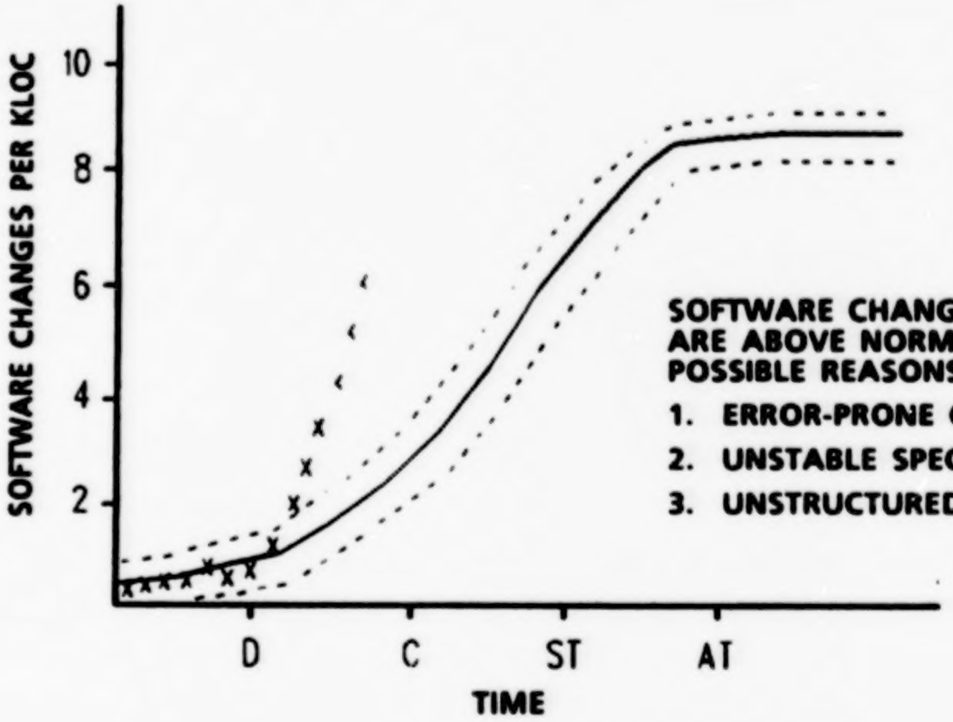
DRO059D A-2

DYNAMITE SCENARIO UTILIZING EXPERT SYSTEM

STEP 1
RETRIEVE DATA FROM DYNAMIC PROJECT FILE

STEP 2
RETRIEVE MODE OF SEL EXPERIENCE

STEP 3
ASSESS MEANING OF COMPARISON



SOFTWARE CHANGES ARE ABOVE NORMAL – POSSIBLE REASONS:

1. ERROR-PRONE CODE – 72%
2. UNSTABLE SPECS – 54%
3. UNSTRUCTURED DEVEL. – 40%

DRO059D A-2

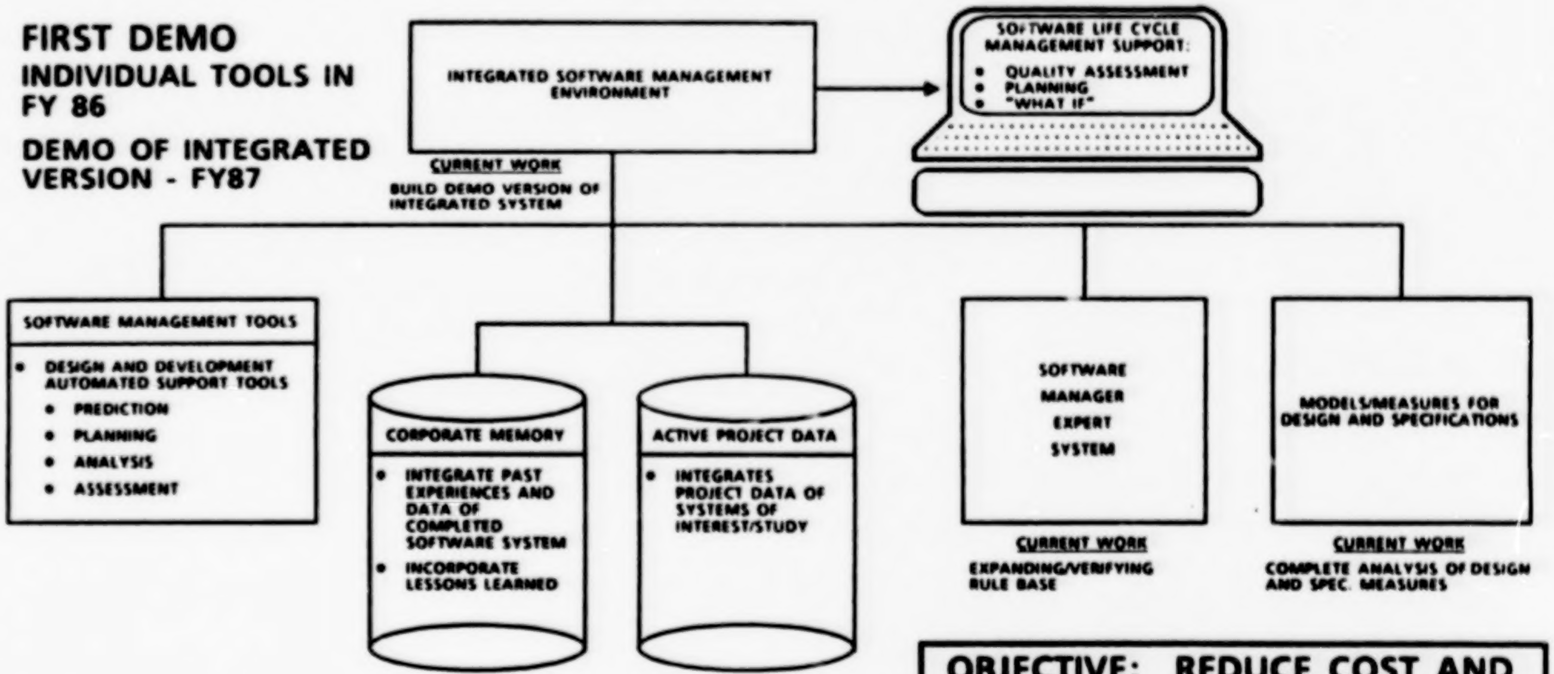
MO&DS
DIRECTORATE
CODE 500

SOFTWARE MANAGEMENT ENVIRONMENT FOR NASA

GSFC

- INTEGRATION OF:
  - ANALYSIS TOOLS FOR MODELING AND MEASURING THE SOFTWARE DEVELOPMENT PROCESS
  - EXPERT SYSTEM FUNCTIONS FOR COMPARISON OF CURRENT PROJECTS TO PAST EXPERIENCE TO PREDICT PROBLEM AREAS

- FIRST DEMO
  INDIVIDUAL TOOLS IN FY 86
- DEMO OF INTEGRATED VERSION - FY87



INTEGRATED SOFTWARE MANAGEMENT ENVIRONMENT

CURRENT WORK
BUILD DEMO VERSION OF INTEGRATED SYSTEM

SOFTWARE LIFE CYCLE MANAGEMENT SUPPORT:
- QUALITY ASSESSMENT
- PLANNING
- "WHAT IF"

SOFTWARE MANAGEMENT TOOLS
- DESIGN AND DEVELOPMENT AUTOMATED SUPPORT TOOLS
  - PREDICTION
  - PLANNING
  - ANALYSIS
  - ASSESSMENT

CORPORATE MEMORY
- INTEGRATE PAST EXPERIENCES AND DATA OF COMPLETED SOFTWARE SYSTEM
- INCORPORATE LESSONS LEARNED

ACTIVE PROJECT DATA
- INTEGRATES PROJECT DATA OF SYSTEMS OF INTEREST/STUDY

SOFTWARE
MANAGER
EXPERT
SYSTEM

CURRENT WORK
EXPANDING/VERIFYING RULE BASE

MODELS/MEASURES FOR DESIGN AND SPECIFICATIONS

CURRENT WORK
COMPLETE ANALYSIS OF DESIGN AND SPEC. MEASURES

OBJECTIVE: REDUCE COST AND IMPROVE RELIABILITY OF NASA SOFTWARE DEVELOPMENT PROJECTS.

DRO059D A-2

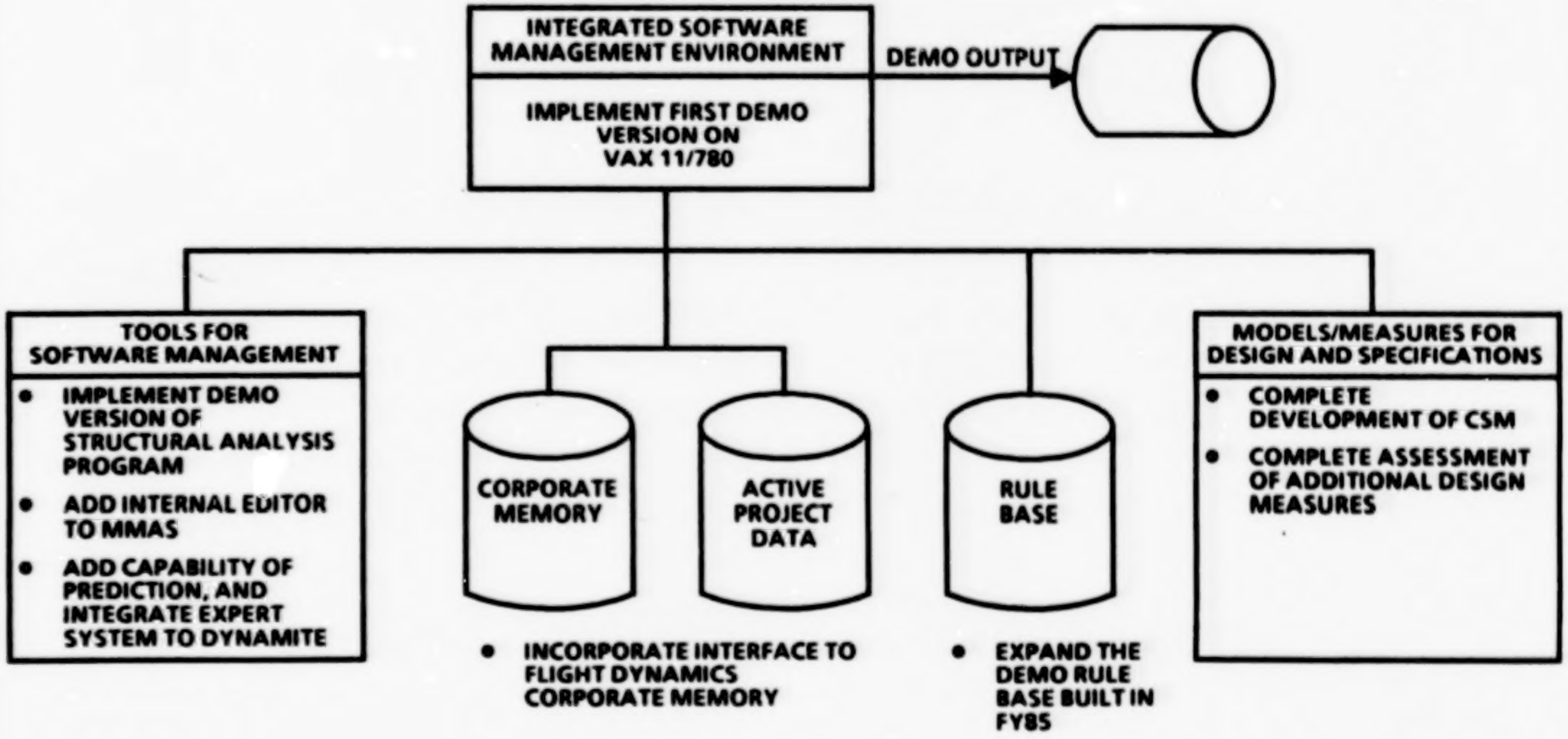
MO&DS DIRECTORATE

CODE 500

SOFTWARE MANAGEMENT ENVIRONMENT FOR NASA

ACCOMPLISHMENTS

FY86



INTEGRATED SOFTWARE MANAGEMENT ENVIRONMENT

- IMPLEMENT FIRST DEMO VERSION ON VAX 11/780

DEMO OUTPUT

TOOLS FOR SOFTWARE MANAGEMENT

- IMPLEMENT DEMO VERSION OF STRUCTURAL ANALYSIS PROGRAM
- ADD INTERNAL EDITOR TO MMAS
- ADD CAPABILITY OF PREDICTION, AND INTEGRATE EXPERT SYSTEM TO DYNAMITE

CORPORATE MEMORY

ACTIVE PROJECT DATA

- INCORPORATE INTERFACE TO FLIGHT DYNAMICS CORPORATE MEMORY

RULE BASE

- EXPAND THE DEMO RULE BASE BUILT IN FY85

MODELS/MEASURES FOR DESIGN AND SPECIFICATIONS

- COMPLETE DEVELOPMENT OF CSM
- COMPLETE ASSESSMENT OF ADDITIONAL DESIGN MEASURES

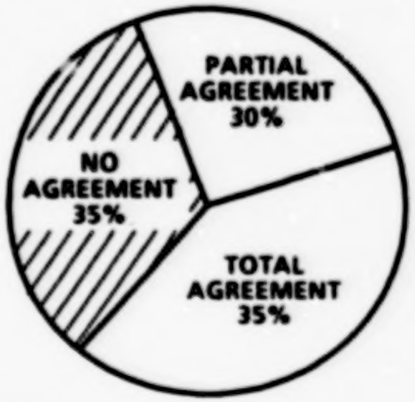
DRO059D A-2

MO&DS DIRECTORATE
CODE 500

SOFTWARE MANAGEMENT ENVIRONMENT FOR NASA

SOFTWARE MANAGEMENT KNOWLEDGE BASE
IS IT FEASIBLE?

- 1986 STUDY OF
  130 RULES USING 3 DIFFERENT METHODS –
  1. TOP DOWN (INDEPENDENTLY – 2 EXPERTS)
  2. BOTTOM UP (INDEPENDENTLY – 2 EXPERTS)
  3. EMPIRICAL STUDIES (SEL DATA BASE AND RELATED STUDIES)
- RESULTS



- 1987 WORK
  - INVESTIGATE DISAGREEMENT
  - DEVELOP NEW RULES
  - REFINE EXISTING RULES

DR0059D A-2

AREAS OF SUPPORT

- CTA
  - ARCHITECT/SYSTEMS ENGINEER INTEGRATED ENVIRONMENT
- CSC
  - BUILD INTEGRATED ENVIRONMENT
- UNIVERSITY OF MARYLAND
  - DEVELOP/ASSESS KNOWLEDGE BASE

SAGA:
An Integrated Software Development Environment

Principle investigator:
Dr. Roy Campbell, U of Illinois

presented by
Kathryn A Smith

NASA Langley Research Center

November 18, 1986

Outline

SAGA System (Software Automation,
Generation and Administration) System

☐ Overview
☐ SAGA System Components
☐ Research Activities
☐ Status at Langley
☐ Plans

Overview

○ SAGA is a system to build software development environments - it is an environment that is used to build other environments

○ An ongoing, large scale project

○ Based on

  ☐ formally defined language forms

  ☐ automatically generated components using those forms

  ☐ other language independent components

○ Aimed towards a fully integrated Software development environment

○ Addresses all phases of the software lifecycle, including management

Research Areas

- Practical organization of the life cycle
- Configuration management
- Software requirements specification
- Executable specifications
- Design methodologies
- Programming
- Verification
- Validation and testing
- Version control
- Maintenance
- Reuse of software libraries
- Documentation
- Integrated environments

SAGA Components

- ○ Fully integrated set of tools - i.e. output from one serves as input to another
- ○ Basically no commercial products - all are done as part of SAGA, currently using some UNIX facilities
  - ☐ little or no proprietary software
  - ☐ research results available
  - ☐ allows full integration of components
  - ☐ provides a vehicle to learn

Current Research Activities

○ **olorin**, a compiler- and editor-generator system to provide syntactic/semantic analysis components of a compiler or editor

○ software management study

  ☐ study of software management techniques at AT&T

  ☐ study of other project management systems

○ **notesfiles**, electronic bulletin board

○ **emake**, extended version of UNIX make utility

○ **maketd**, automatic creation of dependency information for make

SAGA editor (epos)

- Language-oriented incremental, screen editor
- Performs syntactic analysis of edited text
- Uses recognition techniques for syntactic and semantic analysis
- New version, GNU epos, uses GNU Emacs as a user interface to the epos incremental parser
- Currently using MYSTRO (College of William and Mary) to develop language editors

CLEMMA

- Automated configuration librarian prototype, under development
- Goal to provide a means of organizing, indexing,storing and retrieving on-line components of software projects
- Combines the TROLL DBMS (database management system) and the UNIX file system

ENCOMPASS

- ○ Experimental software development environment
- ○ Initial prototype on a Sun workstation under UNIX
- ○ Created using other SAGA components
- ○ Prototype developed for Ada programs
- ○ Includes:
  - ☐ language-oriented editor
  - ☐ test harness
  - ☐ user interface package
- ○ User can invoke IDEAL(Incremental Development Environment for Annotated Languages), an environment in-the-small, with facilities to specify, prototype, test and implement Ada programs

PLEASE(Predicate Logic based ExecutAble SpEcifications)

- ○ Executable specification language
- ○ For requirements verification and rapid prototyping
- ○ Planned features include support of if, while, and assignment statements and support a small, fixed set of data types
- ○ Initial version of ISLET(Incredibly Simple Language oriented Editing Tool), language-oriented editor to create PLEASE specifications and translate them into Ada
- ○ Initial version of software to translate PLEASE specifications into PROLOG procedures and generate support code to call these procedures from Ada

Status at LaRC

- ○ Operational
  - ☐ Notesfiles
  - ☐ emake
- ○ Installed
  - ☐ epos
  - ☐ GNU epos
  - ☐ olorin
  - ☐ Tool support libraries
  - ☐ ENCOMPASS
- ○ Problems due to a compiler bug on the workstation

PLANS

○ Install and evaluate components at Langley as they become available

○ Use ENCOMPASS prototype to examine more advanced features

○ Prototype will be used to examine the practical applications of SAGA

Cognitive-Based Analysis of Software Documentation[1]

Elliot Soloway
Associate Professor of Computer Science
and Psychology
Cognition and Programming Project
(CAPP)
Computer Science Department
Yale University
New Haven, CT 06520

This research was supported in part by the Jet Propulsion Laboratory California Institute of Technology under contract with the National Aeronautics and Space Administration, and by the National Science Foundation grant DCR - 8302382.

---

[1] Presented 11/18/86 by David Littman

TODAY'S TALK

- THE PROBLEM OF DOCUMENTATION IN MAINTENANCE CONTEXT
- SOME OF THE ISSUES WE HAVE STUDIED EMPIRICALLY
- A TASTE OF THE RESULTS OF OUR STUDIES

THE PROBLEM WITH DOCUMENTATION

- CONSTANTLY ENHANCING PROGRAMS
- ENHANCER IS NOT ORIGINAL PROGRAMMER
- BASIC PREMISE OF PROGRAMMERS:
  DON'T TRUST DOCUMENTATION

OUR APPROACH: THE COGNITIVE PERSPECTIVE

- THREE ISSUES IN DOCUMENTATION
    - WHAT TO DOCUMENT?
    - HOW TO DOCUMENT?
    - WHEN TO PRESENT THE INFORMATION?
- OUR ANSWERS TO 3 QUESTIONS ARE BASED ON COGNITIVE PRINCIPLES

OUR METHODOLOGY:
(IN A NUTSHELL)

- **"YOU CAN OBSERVE A LOT BY JUST WATCHING."**
  Yogi Berra
- **"YOU HAVE TO GO DOWN THE ROPE HAND-OVER-HAND."**
  JPL Guru

2 AREAS OF STUDY AT JPL

- AREA 1: EXAMINING THE WHAT QUESTION
    - PROGRAMMER'S USE OF CURRENT DOCUMENTATION
    - CAN WE IMPROVE IT?
    - STUDY DONE IN CONTEXT OF SMALL PROGRAM (300 LINES OF FORTRAN)
    - TASK: MAKE AN ENHANCEMENT

2 AREAS OF STUDY AT JPL (CONTINUED)

- AREA 2: EXAMINING THE WHAT & WHEN QUESTIONS
  - PROGRAMMERS CONFRONT LARGE PROGRAM (60K LINES)
  - 1.5 FEET OF DOCUMENTATION
  - TASK: MAKE AN ENHANCEMENT

AREA 1 STUDY: "WHAT" QUESTION
PHASE I

- METHODOLOGY:
    - GIVE PROGRAMMERS SMALL PROGRAM + STANDARD DOCUMENTATION
    - ASKED THEM TO ENHANCE PROGRAM
    - VIDEOTAPED PROBLEM SOLVING BEHAVIOR
    - RECORDED "TALKING ALOUD" PROTOCOLS
    - 10 JPL PROFESSIONALS

RESULTS: "WHAT" QUESTION
PHASE I

- IDENTIFIED 2 PROGRAM STUDY STRATEGIES USED TO BUILD MENTAL MODEL OF PROGRAM
    - SYSTEMATIC: MENTAL SIMULATION USED TO SEE HOW WHOLE PROGRAM FITS TOGETHER BEFORE ATTEMPTING ENHANCEMENT
    - AS-NEEDED: GO FOR THE PATCH AS SOON AS POSSIBLE
- IF ADOPT SYSTEMATIC STRATEGY ----> CORRECT PATCH
- IF ADOPT AS-NEEDED STRATEGY ----> INCORRECT PATCH
- NO EFFECT AT ALL OF PROGRAMMING EXPERIENCE!!!

DISCUSSION: "WHAT" QUESTION
PHASE I

- SMALL PROGRAM --- SYSTEMATIC STRATEGY POSSIBLE
- BIG PROGRAM --- SYSTEMATIC STRATEGY IMPOSSIBLE
- QUESTIONS:
  - WHAT INFORMATION DID SYSTEMATIC STRATEGY UNCOVER?
  - CAN THAT INFO BE PUT IN DOCUMENTATION?
  - WILL THAT INFO HELP THE AS-NEEDED STRATEGY?

AREA 1 STUDY: "WHAT" QUESTION
PHASE II

- GOAL: FACILITATE AS-NEEDED STRATEGY
- METHOD: PROVIDE KEY INFO IN DOCUMENTATION
  - CAUSAL INTERACTIONS AMONG DELOCALIZED PLANS
- ANOTHER STUDY:
  NEW DOCUMENTATION *vs.* OLD DOCUMENTATION
- PRELIMINARY RESULTS:
  - NEW DOC APPEARED TO HELP AS-NEEDED STRATEGY
  - SUBJECTS WITH MORE EXPERIENCE ESPECIALLY BENEFITTED
  - INTERESTING!!

AREA 2 STUDY:
"WHAT" & "WHEN" QUESTIONS

- CAN WE PROVIDE KEY INFO WHEN IT IS NEEDED?
- KEY INFO: GOALS --- WHAT IS PROGRAM/MODULE SUPPOSED TO DO?
- PROVIDE KEY INFO UP FRONT --- NICELY LABELLED
- LARGE PROGRAM CURRENTLY USED SUCCESSFULLY AT JPL (60K LINES)
- LARGE DOCUMENTATION (1.5 FEET)

METHODOLOGY:

- CURRENT DOC *vs.* REVISED DOC
- REVISION TRIED TO CLARIFY STRUCTURE OF DOCUMENTATION
- 4 JPL PROGRAMMERS:
  (2 ON EACH FORM OF DOCUMENTATION)
- ASK PROGRAMMERS TO MAKE ENHANCEMENT
- ENHANCEMENT ACTUALLY REQUIRED CHANGING ONLY ONE MODULE

RESULTS

- OLD DOC: 2 HOURS TO FIND WHERE TO MAKE CHANGE
- NEW DOC: 20 MINUTES TO FIND WHERE TO MAKE CHANGE (!!!!!)
- BIG DIFFERENCE (BUT ONLY FOUR SUBJECTS ...)

WHY??????

- WE KNOW PROGRAMMERS BUILD MENTAL MODELS OF PROGRAMS
- THE ISSUES HERE ARE:
    - PROGRAMMERS NEED TO BUILD MENTAL MODELS OF DOCUMENTATION
    - HOW CAN WE HELP THEM BUILD MENTAL MODELS OF DOCUMENTATION?

THE POINT IS:

- FOR LARGE PROGRAMS, NEED CLEAR UNDERSTANDING OF STRUCTURE OF DOCUMENTATION
- THAT IS A MENTAL MODEL
- WE SAW THEM BUILD MENTAL MODELS OF DOCUMENTATION:
    - CURRENTLY RELEVANT: "I NEED TO KNOW THIS FOR WHAT I'M DOING NOW"
    - POTENTIALLY RELEVANT: "I MIGHT NEED TO KNOW THIS FOR SOMETHING I DO LATER"
    - IRRELEVANT: "I DO NOT NEED TO KNOW THIS"

LESSONS LEARNED

- BUILDING MENTAL MODEL OF DOC IS
    - IMPORTANT
    - HARD
    - NOT WELL UNDERSTOOD
- NEED TO THINK ABOUT TOOLS THAT SUPPORT BUILDING MENTAL MODELS OF DOCUMENTATION

SUMMARY: SO WHAT? & WHAT NEXT?

- SO WHAT?
  - 2 YEARS HAS YIELDED MUCH
  - CLOSE TO MAKING SPECIFIC PRESCRIPTIONS FOR "WHAT" & "WHEN" QUESTIONS
  - DOCUMENTATION CAN BE IMPROVED AND USEFUL

WHAT NEXT?

- REPLICATE AND EXTEND THE STUDIES
- LOOK AT "WHEN" QUESTION IN MUCH MORE DETAIL
- BUILD COMPUTER-BASED DOCUMENTATION TOOLS
- SYSTEM SUGGESTS WHAT USER MIGHT NEED TO KNOW

MORE WHAT NEXT?

- LOOK AT REUSABILITY
- LIBRARIES OF ROUTINES
- PROBLEM IS HOW TO DOCUMENT THEM?
    - TELL WHAT THE ROUTINE DOES
    - SAY WHAT IS "NON-STANDARD" ABOUT ROUTINE
- MAKE STRUCTURE OF DOCUMENTATION EXPLICIT

WE ARE OPTIMISTIC

- METHODOLOGY IS DEVELOPED
- INITIAL RESULTS
- CLEAR DIRECTION FOR NEXT PHASES OF RESEARCH
- INTEREST OF MANY COMMUNITIES WITH DIVERGENT INTERESTS

SOFTWARE LIFE CYCLE SIMULATION (SLICS) MODELING

JPL

Chi Lin

TASK PARTICIPANTS

Krista Kelly
Merle McKenzie
Debra Synott

JPL

AGENDA

- INTRODUCTION
- SLICS DEVELOPMENT BACKGROUND
- APPROACH
- SLICS
    - SCENARIO OF MODEL APPLICATION
    - INPUTS/OUTPUTS
    - MODEL STRUCTURE
    - EQUATIONS
- ACCOMPLISHMENTS AND FUTURE FOCUS

INTRODUCTION

- OBJECTIVE
    - DEVELOP A MULTI-PURPOSE DYNAMIC SIMULATION MODEL OF THE SOFTWARE LIFE CYCLE PROCESS
- GOAL: PROVIDE SOFTWARE MANAGERS WITH QUALITY PROJECT SIMULATION ON PAR WITH OTHER TECHNICAL SIMULATION TOOLS
    - REDUCE SOFTWARE BUDGETARY AND SCHEDULE PLANNING UNCERTAINTIES
    - PROVIDE STRATEGIC SUPPORT FOR TECHNOLOGY DEVELOPMENT AND PRODUCTIVITY IMPROVEMENT
    - IMPROVE SOFTWARE TASK PLANNING AND MANAGEABILITY
- APPLICATIONS: SUPPORT TRAINING, PLANNING, MONITORING AND CONTROLLING THROUGHOUT THE SOFTWARE LIFE CYCLE

JPL

SLICS DEVELOPMENT BACKGROUND

- CURRENT SOFTWARE SYSTEM CONDITIONS
    - CONTINUING RISE IN SOFTWARE SYSTEM COSTS AND COMPLEXITY
    - CONSTANT CHANGES IN SOFTWARE TECHNOLOGY
    - PERSISTENT COST AND SCHEDULE OVERRUNS
- SOFTWARE LIFE CYCLE ASSESSMENT: THE STATE-OF-THE-ART
    - PICTORIAL MODELS
    - PARAMETRIC COST MODELS
    - STATIC PERT/CPM TOOLS
- SOFTWARE MANAGEMENT NEEDS A MODEL IN WHICH THE DYNAMIC FEEDBACK, NONLINEAR CHARACTERISTICS, AND HUMAN FACTORS OF THE SOFTWARE LIFE CYCLE CAN BE CAPTURED

MODELING APPROACH

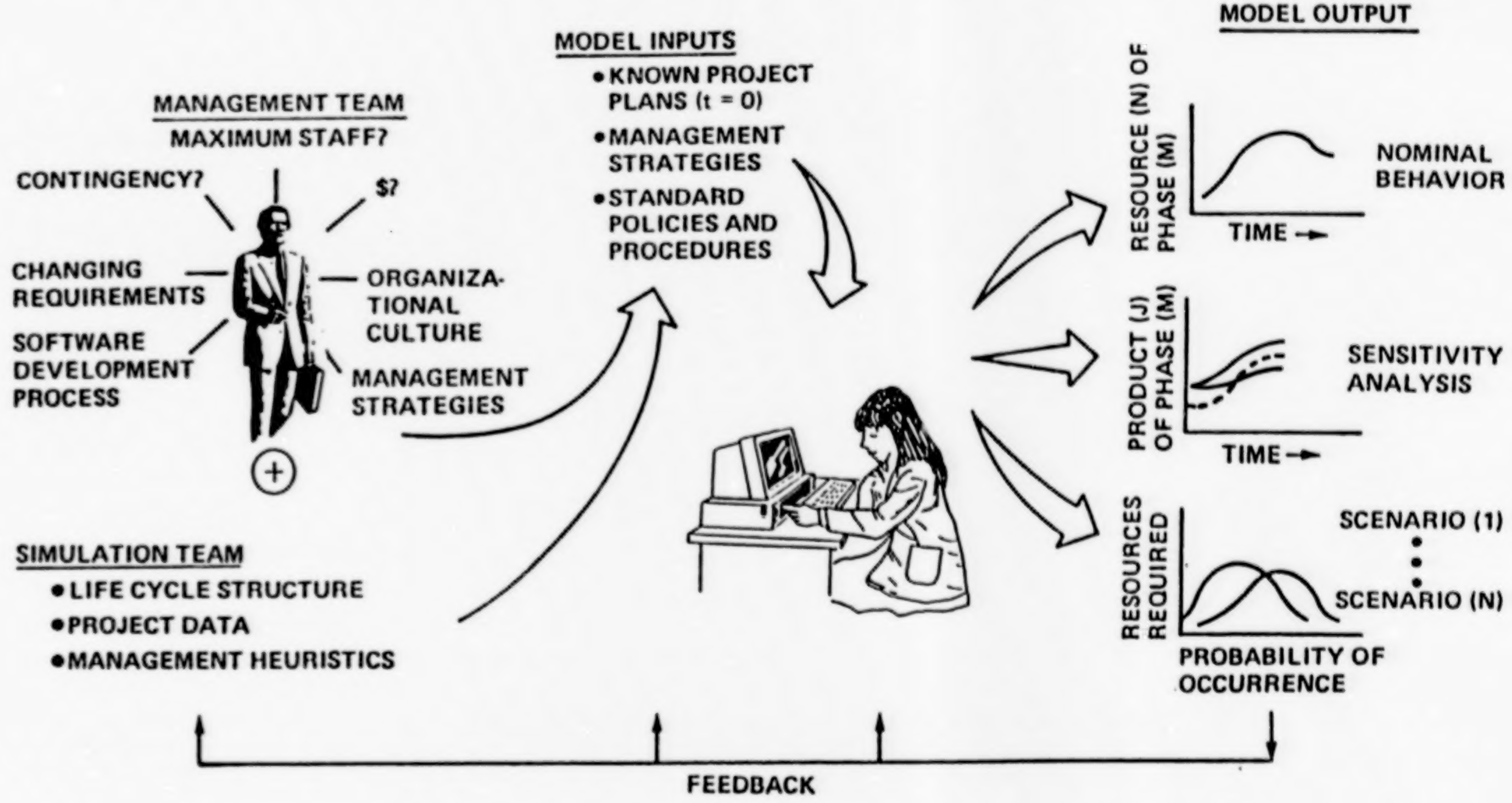
- HYBRID MODELING TECHNIQUE
  - DETAILED SIMULATION
  - DYNAMIC MODELING
- USE HEURISTICS AS WELL AS RECORDED DATA
  - SCHEDULE PRESSURE
  - ADVANCED WORKSTATIONS
  - PROJECT FAMILIARITY
  - HIRING DELAY

JPL

SCENARIO OF MODEL APPLICATION

GOAL
PROVIDE SOFTWARE MANAGERS WITH QUALITY PROJECT SIMULATION ON PAR WITH OTHER TECHNICAL SIMULATION TOOLS

PAYOFF
DYNAMIC CONTROL OF LARGE SOFTWARE PROJECTS



MANAGEMENT TEAM

- MAXIMUM STAFF?
- CONTINGENCY?
- $?
- CHANGING REQUIREMENTS
- ORGANIZATIONAL CULTURE
- SOFTWARE DEVELOPMENT PROCESS
- MANAGEMENT STRATEGIES

SIMULATION TEAM

- LIFE CYCLE STRUCTURE
- PROJECT DATA
- MANAGEMENT HEURISTICS

MODEL INPUTS

- KNOWN PROJECT PLANS (t = 0)
- MANAGEMENT STRATEGIES
- STANDARD POLICIES AND PROCEDURES

MODEL OUTPUT

RESOURCE (N) OF PHASE (M)
TIME →
NOMINAL BEHAVIOR
PRODUCT (J) OF PHASE (M)
TIME →
SENSITIVITY ANALYSIS
RESOURCES REQUIRED
PROBABILITY OF OCCURRENCE
SCENARIO (1)
SCENARIO (N)
FEEDBACK

JPL

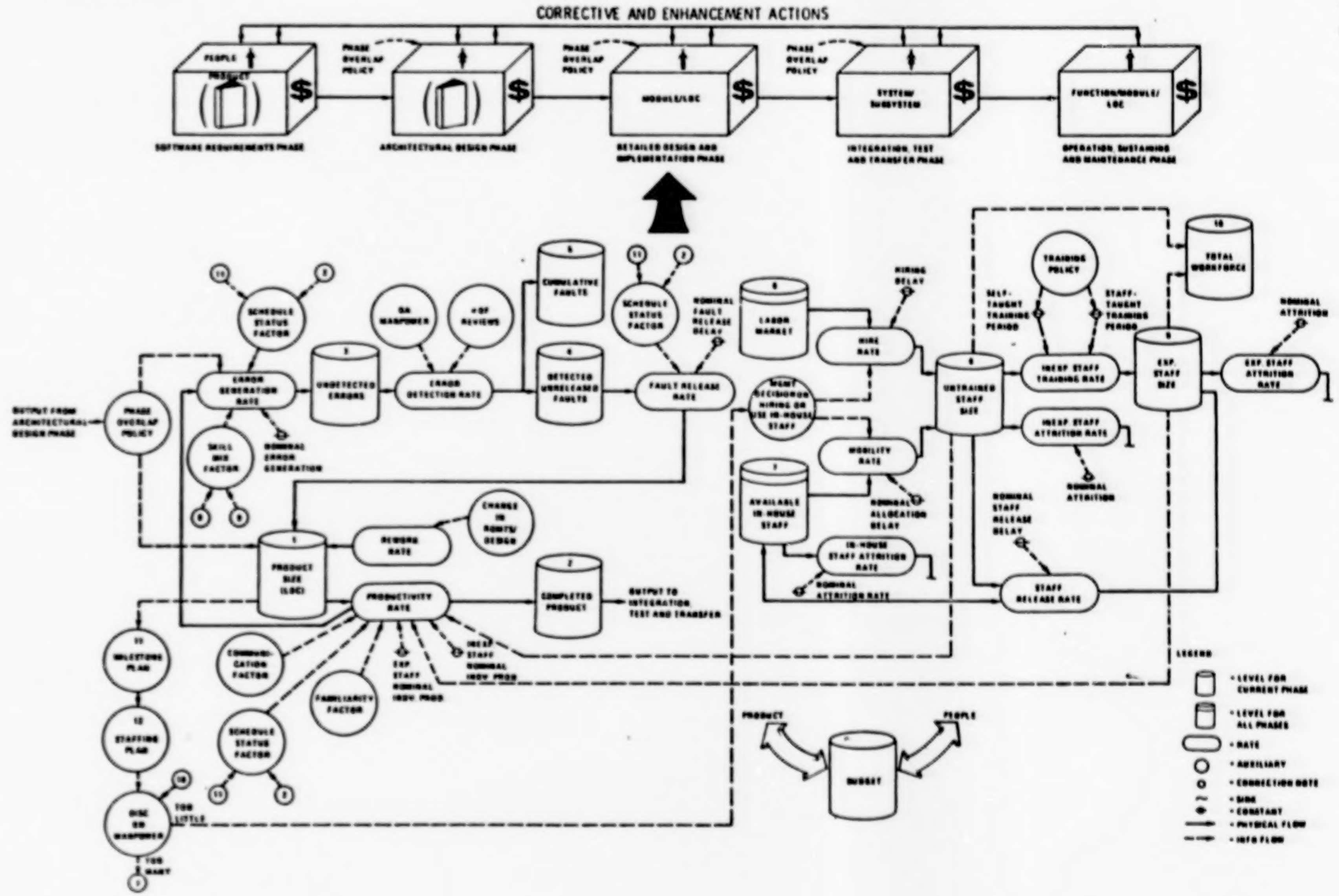
EXAMPLES OF INPUTS AND OUTPUTS

- ESTIMATED INPUTS (AT t = 0)
  - MANDATORY
    - PROJECT SIZE
    - SCHEDULE PLAN
    - COST PER WORKYEAR
    - DEVELOPMENT ENVIRONMENT
    - MANAGEMENT ENVIRONMENT
  - OPTIONAL
    - NOMINAL STAFF PRODUCTIVITY
    - PROJECT CONSTRAINTS
      - MAXIMUM STAFF
      - MAXIMUM BUDGET
      - MINIMUM FUNCTIONALITY

- OUTPUTS AS A FUNCTION OF TIME
  - ACTUAL STAFF PROFILE
  - TOTAL PHASE AND PROJECT COSTS
  - ACTUAL SCHEDULE COMPLETION TIMES
  - TOTAL REWORK RATE
  - THE IMPACT OF CHANGES IN REQUIREMENTS ON SCHEDULE, COST, AND STAFF PRODUCTIVITY
  - PHASE OVERLAP IMPACT ON SCHEDULE, COST, AND STAFF PROFILE
  - VARYING PRODUCTIVITY RATE

ORIGINAL PAGE IS
OF POOR QUALITY

SLICS MODEL STRUCTURE


CORRECTIVE AND ENHANCEMENT ACTIONS
PEOPLE
PRODUCT
PHASE OVERLAP POLICY
MODULE/LOC
SYSTEM/ SUBSYSTEM
FUNCTION/MODULE/ LOC
$
SOFTWARE REQUIREMENTS PHASE
ARCHITECTURAL DESIGN PHASE
DETAILED DESIGN AND IMPLEMENTATION PHASE
INTEGRATION, TEST AND TRANSFER PHASE
OPERATION, SUSTAINING AND MAINTENANCE PHASE
OUTPUT FROM ARCHITECTURAL DESIGN PHASE
PHASE OVERLAP POLICY
SCHEDULE STATUS FACTOR
11
2
ERROR GENERATION RATE
SKILL MIX FACTOR
8
9
NOMINAL ERROR GENERATION
3
UNDETECTED ERRORS
QA MANPOWER
# OF REVIEWS
ERROR DETECTION RATE
5
CUMULATIVE FAULTS
4
DETECTED UNRELEASED FAULTS
NOMINAL FAULT RELEASE DELAY
FAULT RELEASE RATE
1
PRODUCT SIZE (LOC)
REWORK RATE
CHANGE IN RQMTS/ DESIGN
PRODUCTIVITY RATE
2
COMPLETED PRODUCT
OUTPUT TO INTEGRATION, TEST AND TRANSFER
COMMUNICATION FACTOR
FAMILIARITY FACTOR
EXP STAFF NOMINAL INDV PROD
INEXP STAFF NOMINAL INDV PROD
11
MILESTONE PLAN
12
STAFFING PLAN
DISC ON MANPOWER
10
TOO LITTLE
TOO MANY
7
6
LABOR MARKET
HIRING DELAY
HIRE RATE
MGMT DECISION ON HIRING OR USE IN-HOUSE STAFF
MOBILITY RATE
NOMINAL ALLOCATION DELAY
7
AVAILABLE IN-HOUSE STAFF
IN-HOUSE STAFF ATTRITION RATE
NOMINAL ATTRITION RATE
8
UNTRAINED STAFF SIZE
TRAINING POLICY
SELF-TAUGHT TRAINING PERIOD
STAFF-TAUGHT TRAINING PERIOD
INEXP STAFF TRAINING RATE
INEXP STAFF ATTRITION RATE
NOMINAL ATTRITION
NOMINAL STAFF RELEASE DELAY
STAFF RELEASE RATE
9
EXP. STAFF SIZE
EXP STAFF ATTRITION RATE
NOMINAL ATTRITION
10
TOTAL WORKFORCE
PRODUCT
PEOPLE
BUDGET
LEGEND
= LEVEL FOR CURRENT PHASE
= LEVEL FOR ALL PHASES
= RATE
= AUXILIARY
= CONNECTION NOTE
= SINK
= CONSTANT
= PHYSICAL FLOW
= INFO FLOW

JPL

EXAMPLE OF EQUATIONS

$$\frac{d}{dt}\,\text{WORKLOAD}_i = \text{NEWWORK}_i + \sum_i^n \text{REWORK}_i - \text{PRODUCTION}_i$$

INITIAL VALUE $\text{WORKLOAD}_i\ (t = 0) = 0$

$$\text{NEWWORK}_i = \begin{cases} 0 & \text{WHEN STATUS}_{(i-1)} \leq X\% \\ \text{PULSE (EST. WORKLOAD}_i) & \text{OTHERWISE} \end{cases}$$

$\text{REWORK}_i = \text{DELAY}_i\ (\text{ERROR FOUND})_i$

$\text{PRODUCTION}_i = S_i\ P_i\ P_\gamma\ P_\beta \ldots P_\sigma$

S_i : STAFF SIZE

P_i : STAFF PRODUCTIVITY UNDER NOMINAL CONDITIONS

E.G., P_γ: SCHEDULE PRESSURE FACTOR

$$P_\gamma = \begin{cases} 1 & \text{WHEN PROJECT STATUS IS NORMAL} \\ 0 < \text{HEURISTIC TABLE (STATUS}_i) < 1 & \text{OTHERWISE} \end{cases}$$

JPL ACCOMPLISHMENTS AND FUTURE FOCUS

- MAJOR ACCOMPLISHMENTS
  - DEVELOPED PROTOTYPE MODEL (SLICS-JPL)
  - VALIDATED APPROACH: APPLIED TO A SHUTTLE PROJECT (SLICS-STAR*)
    - STRUCTURE
    - DATA AND HEURISTICS
    - SIMULATION OUTPUT
  - INITIATED TECHNOLOGY TRANSFER TO SPACE STATION
    - PRELIMINARY STRUCTURE FOR THE SLICS-SSE TESTBED
    - PRELIMINARY MANAGEMENT INPUT/OUTPUT
  - PURSUED FURTHER TECHNOLOGY TRANSFER TO CODES E AND T
- FUTURE FOCUS
  - CORE TECHNOLOGY RESEARCH
    - MULTI-LIFE CYCLE SIMULATOR
    - EXPERT SYSTEM FRONT-END
    - AI AND EXPERT SYSTEM LIFE CYCLE
  - APPLICATIONS
    - COMPLETE SSE TESTBED MODEL
    - APPLY TECHNOLOGY TO CODE E, S, T

*SPACE TRANSPORTATION AUTOMATED RECONFIGURATION

SOFTWARE ENGINEERING WITH REUSABLE COMPONENTS TASK

M. J. Steinbacher

JPL

November 18, 1986

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

C0038
A0017

11-18-86

JPL SOFTWARE ENGINEERING WITH COMPONENTS

- Components include source code, design, test cases, and other software products
- Purpose of this task is to use past and current efforts within DoD (STARS, SEI, ---), NASA, and Industry to investigate software engineering with components as a means of improving quality and productivity while decreasing life cycle costs

C0039
A0017

11-18-86

JPL SOFTWARE ENGINEERING WITH COMPONENTS

Key technology areas

- Guidelines for the development of/evaluation and certification criteria for components
- Techniques and methods for the development of software systems out of components, and software systems partitioning and design techniques and methods to facilitate the identification and development of reusable components
- Library technology advancement (and interface with)

C0040
A0017

11-18-86

JPL APPROACH TO TECHNOLOGY INVESTIGATION

For component technology

- Input
  - STARS — Application Area: reusability guidebook and CAMP project
  - STARS — Measurement Area metrics work
  - GSFC — SEL metrics work and Ada guidelines
  - Yale studies, reports and guidelines
  - DoD — Ada repository
  - U of H/JSC, other Centers, and Industry reports
  - JPL — Reports and Ada components
- Process
  - Develop an evaluation criterion from above input; collect existing components and evaluate against criteria; update criteria and produce a guideline for creating reusable components
- Products
  - Evaluation criteria; development guidelines; a repository of "evaluated" components

C0041
A0017

11-18-86

APPROACH TO TECHNOLOGY INVESTIGATION (Cont'd)

For software system technology

- Input
    - Components, experience, etc. — from component work
    - STARS, Application Area — CAMP Project Experiences
    - SEI — Process Architecture Project and Reuse Project
    - Yale studies
    - Design/Partitioning of several current software systems
- Process
    - Using components available, investigate various approaches to develop a software system out of components. Investigate how different design/partitioning approaches affect ability to identify and locate parts. Compare findings to design/partitioning of several existing systems

APPROACH TO TECHNOLOGY INVESTIGATION (Cont'd)

Long term

- Develop a strategy for the development of software systems using components and a ''software architect assistant''

SOFTWARE ENGINEERING WITH REUSABLE COMPONENTS
ATTACHMENT

JPL

KEY TECHNOLOGY AREAS

Component Technology Issues Examples:

- o What is the criteria for determining "reusability" of a component?
  - reliability, maintainability, other "ility" measures/metrics
  - encapsulation
  - commonality with similar or different application areas
  - documentation and other related software products
- o What are the behavorial/context/environmental factors?
  - side effects or singularities
  - target, word size, etc. constraints
  - operating conditions constraints
- o What are attributes for determining the confidence factor (trust level) of a component?
  - what other information do users need/want to know in order to "trust" a component -- related to cognitive process studies and the need to know some "why's" and "what's", not just "how's"
- o What are the guidelines for developing reusable components?
  - use of data abstraction, generics, typing mechanisms
  - encapsulation and packaging
  - data management and other hardware/operating system considerations
  - trade between generality and specificity
  - information and related data products needed to support reuse

Software System Technology Issues Examples:

- o What is the affect on the requirements and design phases when component reuse is desired?
  - requirements/design phases iteration and interaction
  - affect on design phases when both requirements and parts are input
- o What approaches/techniques can be employed during the design phase to incorporate the proper balance of systems-driven and parts-driven strategies?

Learning the Behavior of Software Systems From Executable Specifications

Christian Wild

Associate Professor

Old Dominion University

Norfolk, VA

* Supported by NASA Grant 1-439

Objective

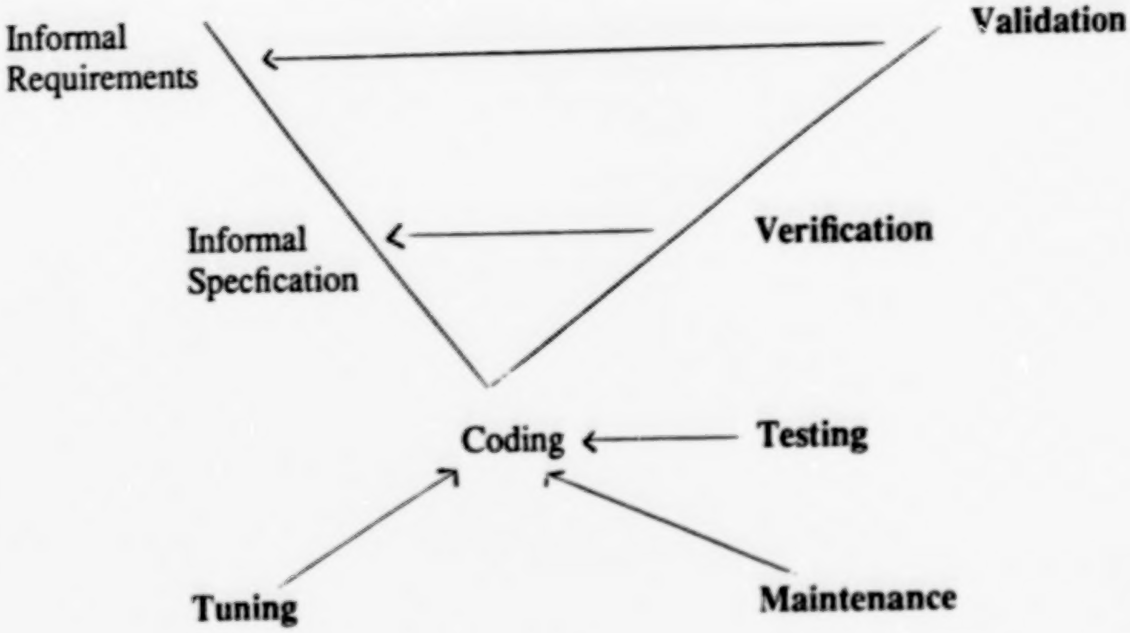
Significantly Increase the **Reliability** and **Robustness** of Software Systems

- Reliability - Development
- Robustness - Operation

Statement of Problem

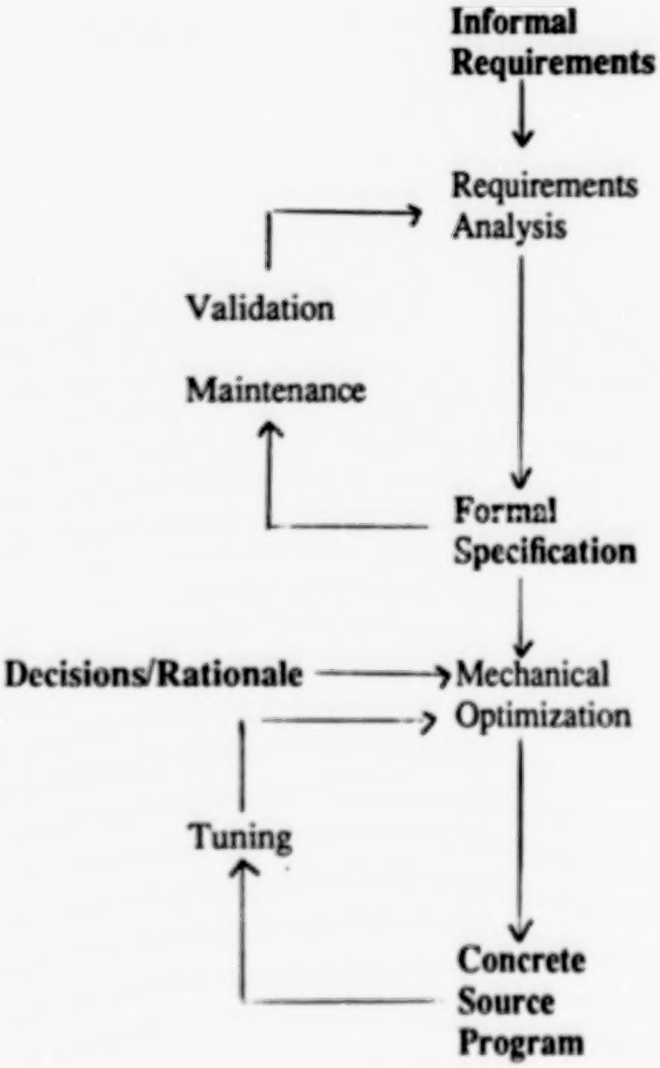
- Complexity increasing
- Criticality increasing
- First implies automation
- Second implies formal analysis
- which in turn implies formal description
- Orders of magnitude increase in reliability cannot be met by traditional software engineering approaches.
- more than just integrated tools at the current level of sophitication. (not smart editors and command interpretors)

Current Software Development Paradigm



Informal
Requirements
Validation
Informal
Specfication
Verification
Coding
Testing
Tuning
Maintenance

Automation Based Software Development Paradigm



Informal Requirements

Requirements Analysis

Validation

Maintenance

Formal Specification

Decisions/Rationale

Mechanical Optimization

Tuning

Concrete Source Program

- **Slide adapted from ISI at USC**

→ Symbolic Execution

↓

Set of Behaviors

↓

Generate Hypotheses

Generalize Hypotheses

↓

Hypotheses of Behavior

↓

Theorem Prover

↓

Assertions

Supporting Technologies

- Formal Specification Languages
- Program Verification
- Automatic Programming
- Rapid Prototyping
- Machine Learning
- Automated Reasoning
- Knowledge-Based Systems.
- Software Fault Tolerance

Definitions

- **Behavior** - Result of applying a function
- **Formal Specification** - Implicitly defines behavior
- **Set of Behaviors** - Explicitly defines behavior
- **State** - Sequence of function calls which generated a data object
- **Canonical Form** - Simplest sequence of function calls which exhibits equivalent behavior

Formal Specification for Queues

Type Queue(Integer)

SYNTAX

Newq -> Queue
Addq(Queue,Integer) -> Queue
Deleteq(Queue) -> Queue
Frontq(Queue) -> Integer ∪ {error}
Isemptyq(Queue) -> Boolean

SEMANTICS
For all q : Queue; i : integer Let

1) Isemptyq(Newq) = True
2) Isemptyq(Addq(q,i)) = False
3) Frontq(Newq) = error
4) Frontq(Addq(q,i)) = If Isemptyq(q) then i
else Frontq(q)
5) Deleteq(Newq) = Newq
6) Deleteq(Addq(q,i)) = If Isemptyq(q) then Newq
else Addq(Deleteq(q),i)

End Queue

Examples of Behavior

Generators:

ADDQ(ADDQ(NEWQ,I_1),I_2)

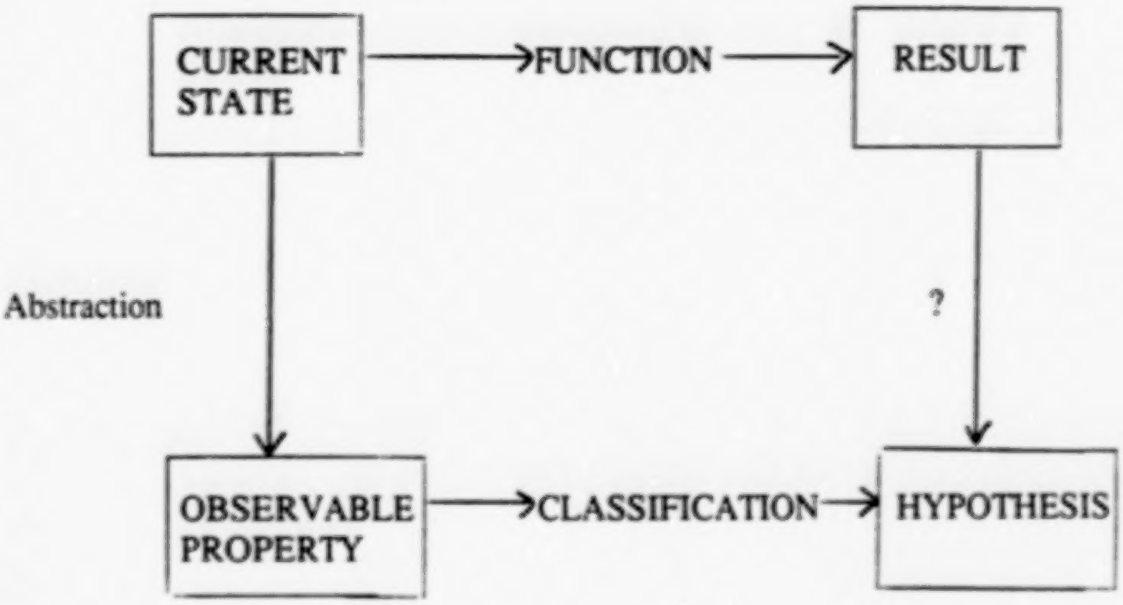
DELETEQ(ADDQ(ADDQ(NEWQ,I_1),I_2)) = ADDQ(NEWQ,I_2)

Behaviors:

ISEMPTYQ(DELETEQ(ADDQ(ADDQ(NEWQ,I_1),I_2))) = False

FRONTQ(DELETEQ(ADDQ(ADDQ(NEWQ,I_1),I_2))) = I_2

Learning Hypothesis of Behavior


CURRENT STATE
FUNCTION
RESULT
Abstraction
?
OBSERVABLE PROPERTY
CLASSIFICATION
HYPOTHESIS

Hypotheses are of the form:

$\text{FUNCTION}(X) = \text{RESULT} \ ? \ \text{A\_F}(X) \in \{\text{PVs}\}$

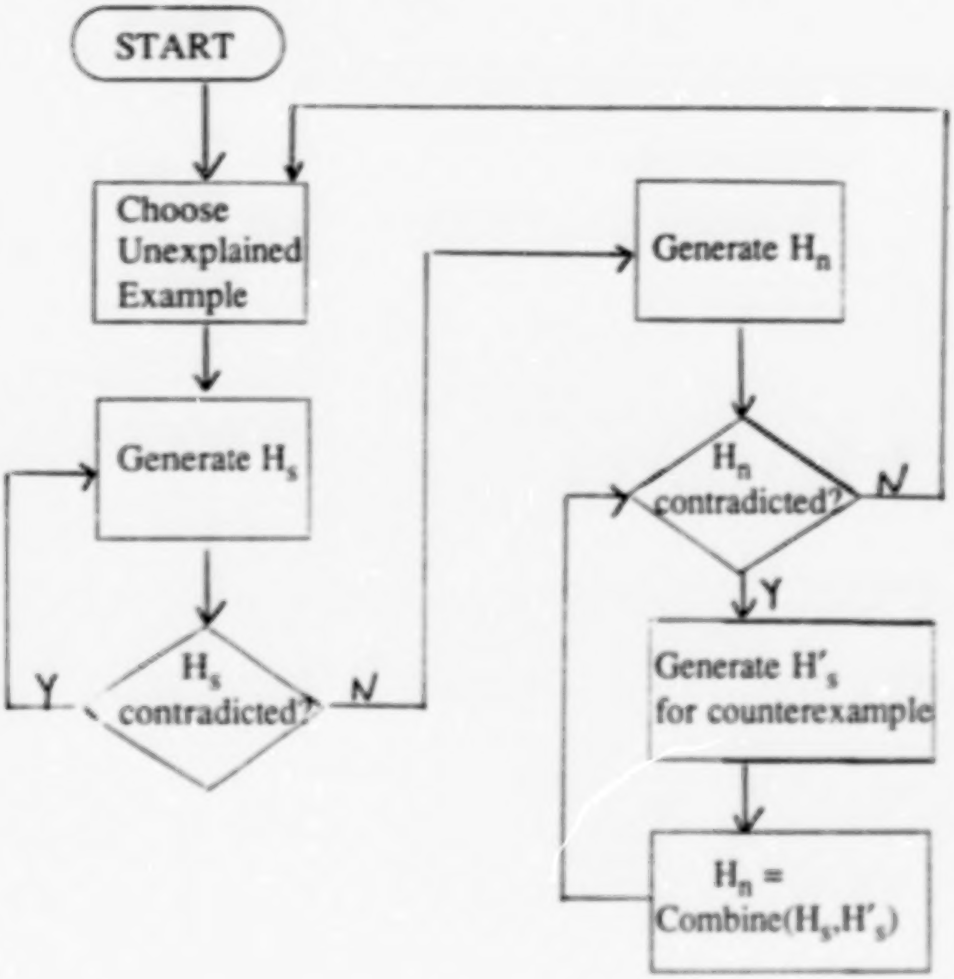
Where ? is:

IF - sufficient

ONLY IF - necessary

IFF - necessary and sufficient

Learning Algorithm


START
Choose Unexplained Example
Generate H_s
H_s contradicted?
Y
N
Generate H_n
H_n contradicted?
N
Y
Generate H'_s for counterexample
H_n = Combine(H_s,H'_s)

Nomenclature

- The AF is of the form #*X*, where X is A (D or N) if the AF counts the number of *Addqs* (*Deleteqs* or *Newqs*) in the WFF.
- AFs on the canonical form will be designated by a adding a prime (').
- AFs which count events occurring before a key event are denoted by a subscript of pre/I_n, where I_n is the item added at the key event.
- For events occurring after the key event the subscript $post/I_n$ is used.

Deleteq(Addq(Addq(NewqJ_1)J_2)) : ($\#A=2, \#A'=1, \#N=1, \#A_{pre/I_2}=1, \#A_{post/I_1}=1$)

Isemptyq(Newq) = T : (#N=1)
$H1_s$ Isemptyq(Q) = T if #N $\in \{1\}$
Isemptyq(Addq(Newq,J_1)) = F: (#A'=1,#N=1,#A=1)

DIFF : {#A',#A}
OUT : {#N}

$H2_s$ Isemptyq(Q) = T if #A' $\in \{0\}$
$H2_n$ Isemptyq(Q) = T only if #A' $\in \{0\}$

Isemptyq(Addq(Newq,J_1)) = F: (#A'=1,#N=1,#A=1)
  $H3_s$) Isemptyq(Q) = F if #A' member-of {1}
  $H3_n$) Isemptyq(Q) = F only if #A' member-of {1}

Isemptyq(Addq(Addq(Newq,J_1),J_2)) = F: (#A'=2)
  $H4_s$) Isemptyq(Q) = F if #A' member-of {2}
  $H5_s$) Isemptyq(Q) = F if #A' member-of {1,2}
  $H5_n$) Isemptyq(Q) = F only if #A' member-of {1,2}

  H6) Isemptyq(Q) = F iff #A' member-of {1..∞}

Program Synthesis for *Isemptyq*

```
IF #A' = 0 THEN TRUE
ELSE FALSE
```

Demands which this behavior places on generators:

NEWQ:	Set #A' to 0
ADDQ:	Increment #A' by 1
DELETEQ:	Decrement #A' by 1

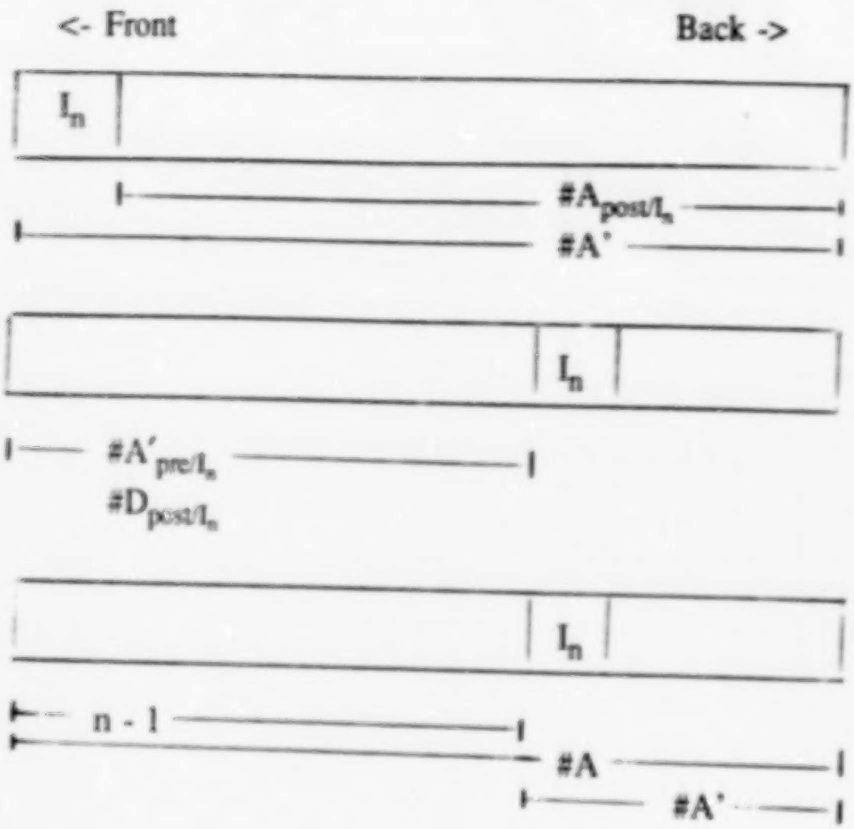
More Complicated Result

Raw Output:

H10) Frontq(Q) = I_2 if $(\#A', \#A_{post/I_2}) \in \{(1,0),(2,1),(3,2),...\}$
H11) Frontq(Q) = I_2 if $(\#D_{post/I_2}, \#A'_{pre/I_2}) \in \{(0,0),(1,1)\}$
H12) Frontq(Q) = I_2 if $(\#A, \#A') \in \{(2,1),(3,2),(4,3),...\}$

Generalized Output:

H10'') Frontq(Q) = I_n iff $(\#A' - \#A_{post/I_n} = 1)$
H11'') Frontq(Q) = I_n iff $(\#D_{post/I_n} - \#A'_{pre/I_n} = 0)$
H12'') Frontq(Q) = I_n iff $(\#A - \#A' = n-1)$



<- Front
Back ->
I_n
$\#A_{post/I_n}$
$\#A'$
I_n
$\#A'_{pre/I_n}$
$\#D_{post/I_n}$
I_n
n - 1
$\#A$
$\#A'$

Conclusions

- Machine Analysis of Executable Specifications:
    - Program Synthesis
    - Automated Test Case Generation
    - Assertion Generation for Run-Time Monitoring
    - Feedback to Requirements Analysis
- Symbiotic Interaction:
    - Machine Learning
    - Symbolic Execution
    - Theorem Proving

NASA MISSION SUPPORT DIRECTORATE JSC

JSC SPONSORED RESEARCH ACTIVITY
UNIVERSITY OF HOUSTON AT CLEAR LAKE

NOVEMBER 18, 1986
STEVE GORMAN / JSC FR12

NASA
National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston Texas

JSC FORM 547 (JUNE 86)

OUTLINE

1. PRE RICIS (MAY 1986) UHCL RESEARCH ACTIVITY
2. FORMATION OF RICIS COOPERATIVE AGREEMENT
3. RICIS CHARTER - GENERAL AREAS OF RESEARCH
4. ESTABLISHED TASKS
5. PLANNED ACTIVITY
6. SUMMARY AND DISCUSSION

PRE RICIS UHCL RESEARCH ACTIVITY

- NUMEROUS TASKS AT UHCL
  - Ada BETA TEST SITE
  - DATA BASE STUDIES
  - ARTIFICIAL INTELLIGENCE STUDIES
- NO CENTRAL COORDINATION
- MANY PROBLEMS AROSE IN MANAGING THE CONTRACTS
  - FUNDING LIMITS EXCEEDED
  - SOLE SOURCE JUSTIFICATIONS ETC.
  - COSTLY AND TIME COMSUMING PAPER MILL
- PERCEPTION BY JSC COMMUNITY THAT THE SPACE CENTER AND THE UNIVERSITY SHOULD WORK MORE CLOSELY TOGETHER

THE RESEARCH INSTITUTE FOR COMPUTING AND INFORMATION SYSTEMS

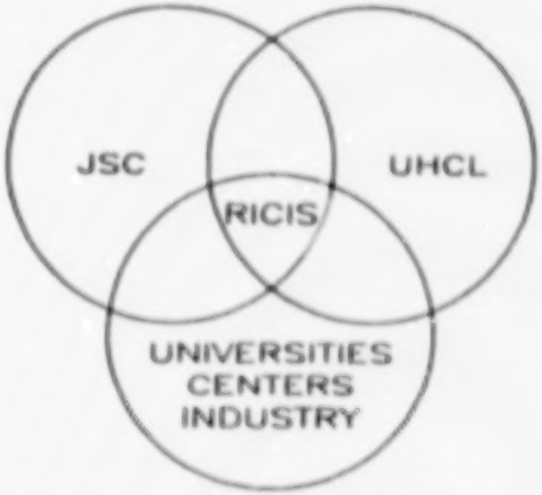
- "TOM AND JERRY" ACCORDS SIGNED IN 1983 (TOM STAUFFER, UHCL AND JERRY GRIFFEN, JSC)
- TASK FORCE FORMED TO EXPLORE POTENTIAL AREAS OF COOPERATION
- RICIS CONCEPT GREW OUT OF TASK FORCE ACTIVITIES
- COOPERATIVE AGREEMENT BETWEEN NASA/JSC AND THE UNIVERSITY OF HOUSTON AT CLEAR LAKE
- ESTABLISHED MAY 1986
- TO BE RENEWED YEARLY
- ACTIVE UHCL RESEARCH CONTRACTS WERE FOLDED INTO RICIS
- GOAL:

  TO IDENTIFY, DEVELOP, APPLY AND EVALUATE ADVANCEMENTS IN COMPUTING AND INFORMATION TECHNOLOGY REQUIRED IN THE CONDUCT OF NASA/JSC'S VARIOUS MISSIONS

GENERAL AREAS OF RESEARCH

- COMPUTER SYSTEMS AND SOFTWARE ENGINEERING
- ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS
- MATHEMATICAL AND STATISTICAL ANALYSIS
- INFORMATION MANAGEMENT
- EDUCATION AND TRAINING

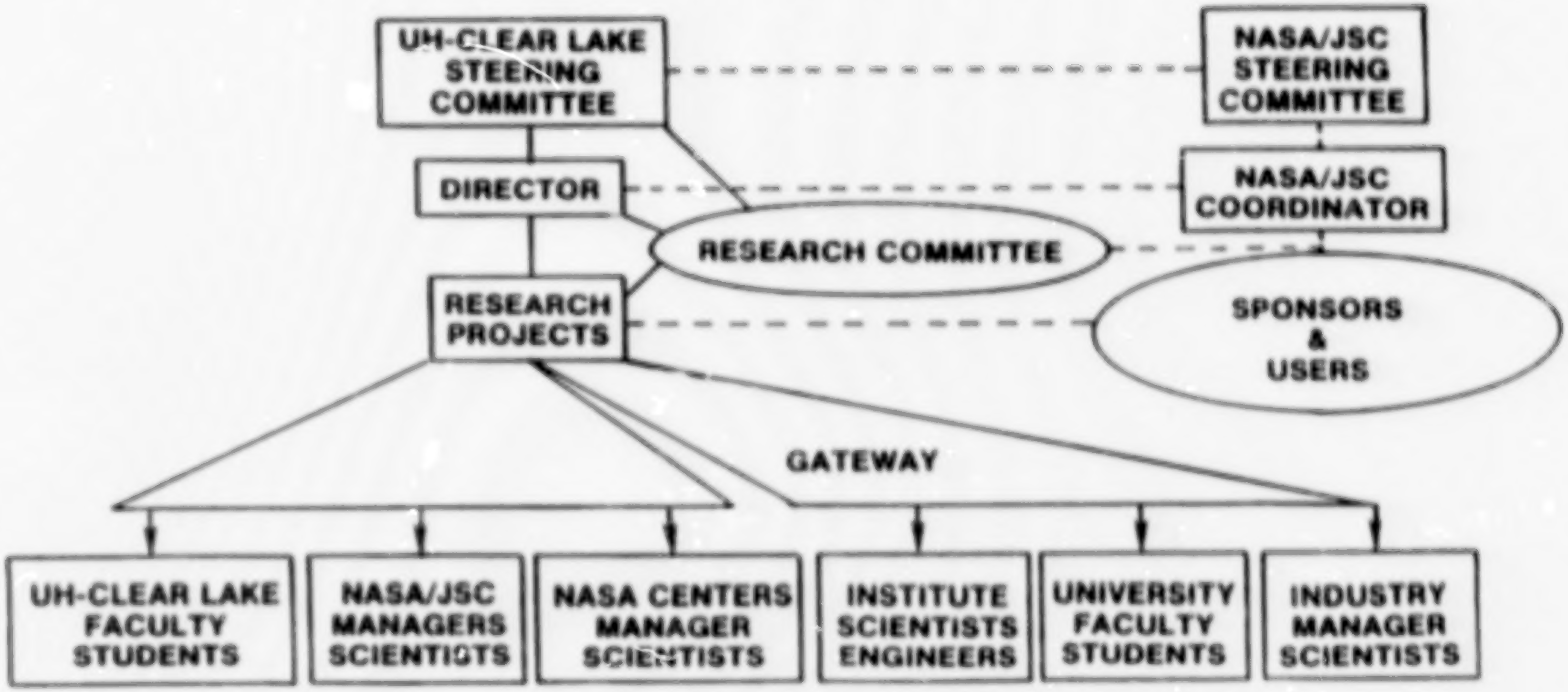
RICIS PARTICIPANTS


JSC
UHCL
RICIS
UNIVERSITIES
CENTERS
INDUSTRY

UNIVERSITY OF HOUSTON AT CLEAR LAKE
NASA/JSC
RESEARCH INSTITUTES AND CENTERS
INDUSTRY
OTHER UNIVERSITIES AND NASA CENTERS

RESEARCH INSTITUTE
for
COMPUTING and INFORMATION SYSTEMS

MANAGEMENT STRUCTURE



UH-CLEAR LAKE STEERING COMMITTEE

NASA/JSC STEERING COMMITTEE

DIRECTOR

NASA/JSC COORDINATOR

RESEARCH COMMITTEE

RESEARCH PROJECTS

SPONSORS & USERS

GATEWAY

UH-CLEAR LAKE FACULTY STUDENTS

NASA/JSC MANAGERS SCIENTISTS

NASA CENTERS MANAGER SCIENTISTS

INSTITUTE SCIENTISTS ENGINEERS

UNIVERSITY FACULTY STUDENTS

INDUSTRY MANAGER SCIENTISTS

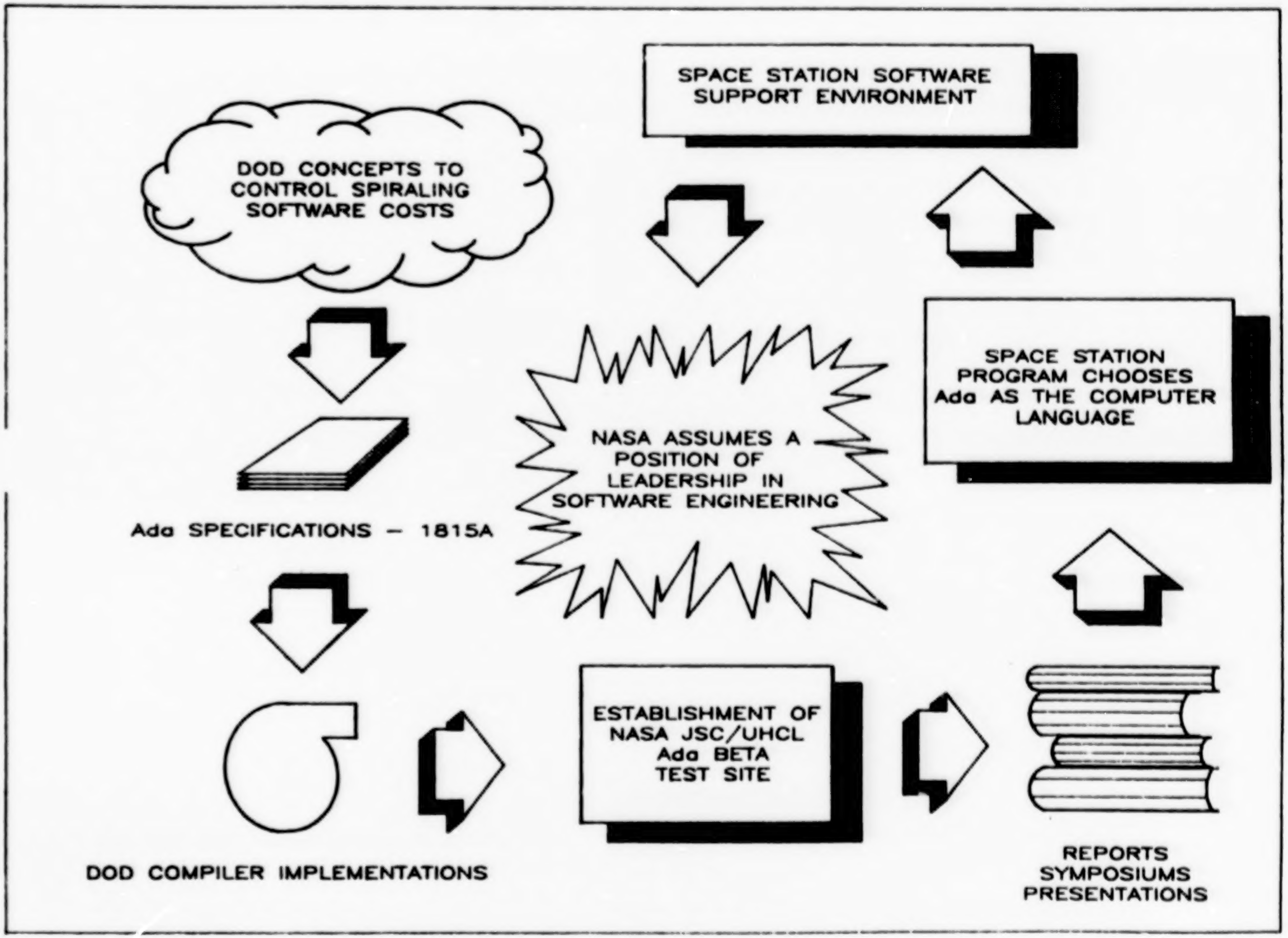
ESTABLISHED TASKS
COMPUTER SYSTEMS AND SOFTWARE ENGINEERING

- Ada BETA TEST SITE
- JOINT NASA/JSC SOFTWARE ENGINEERING RESEARCH CENTER (SERC)
- DEVELOPMENT OF A PROOF-OF-CONCEPT PROTOTYPE OF THE CLEAR LAKE MODEL FOR Ada RUN TIME SUPPORT ENVIRONMENTS
- CONVERSION OF STATE-OF-THE-ART EXPERT SYSTEM SHELL (ART) TO Ada
- BENCHMARKING Ada ON EMBEDDED COMPUTER SYSTEMS
- Ada RELATED INFORMATION DATABASE
- DEVELOPMENT OF DMS TESTBED USERS MANUAL
- SUPPORT TO NASA AND SERC ON Ada RELATED BOARDS AND COMMITTEES

C-2

Ada BETA TEST SITE

- RTOP ACTIVITY - NOW COMPLETE, FUNDED 1983 - 1986
- ACCOMPLISHMENTS:
  - TECHNOLOGY TRANSFER - STATION CHOOSES Ada
  - SUCCESSFUL SYMPOSIUMS
    - MINI-SYMPOSIUMS, ATOP CONTRACTORS & OTHERS
    - FIRST INTERNATIONAL CONFERENCE ON Ada PROGRAMMING APPLICATIONS FOR THE SPACE STATION
  - MINIMUM APSE TOOLSET ANALYSIS
  - BENCHMARKING OF Ada IMPLEMENTATIONS
  - INFLUENCE ON HIGH Ada COUNCILS AND COMMITTEES
    - SIGAda Ada RUN TIME ENVIRONMENT WORKING GROUP
    - CAIS WORKING GROUP
    - Ada BOARD
    - KERNAL INTERFACE TEAM (KIT)
  - ESTABLISHMENT OF SIMTEL LIBRARY AT JSC AND UHCL
  - EXTENSIVE PARTICIPATION IN ATOP ACTIVITY BY THE AEROSPACE AND COMPUTING COMMUNITY
- UHCL CONTACT: DR. CHARLES McKAY



DOD CONCEPTS TO CONTROL SPIRALING SOFTWARE COSTS

Ada SPECIFICATIONS – 1815A

DOD COMPILER IMPLEMENTATIONS

ESTABLISHMENT OF NASA JSC/UHCL Ada BETA TEST SITE

REPORTS
SYMPOSIUMS
PRESENTATIONS

SPACE STATION PROGRAM CHOOSES Ada AS THE COMPUTER LANGUAGE

SPACE STATION SOFTWARE SUPPORT ENVIRONMENT

NASA ASSUMES A POSITION OF LEADERSHIP IN SOFTWARE ENGINEERING

NASA JSC/UHCL SOFTWARE ENGINEERING RESEARCH CENTER

- FOLLOW ON TO THE Ada BETA TEST SITE RTOP
- EXPAND FOCUS TO COVER ENTIRE SOFTWARE LIFE CYCLE
  - DOD 2167 MODEL AS A STARTING POINT
  - SEEK TECHNOLOGY TRANSFER IN:
    - REQUIREMENTS ANALYSIS
    - DESIGN
    - CODE
    - VERIFICATION AND VALIDATION
    - SUSTAINING ENGINEERING
    - SYSTEM RELIABLILTY AND QUALITY ASSURANCE
    - COMPUTER SYSTEM SECURITY
- PARTICULAR EMPHASIS ON DISTRIBUTED COMPUTING ISSUES
- AFFILIATION WITH OTHER CENTERS AND INSTITUTES
    - SOFTWARE ENGINEERING INSTITUTE
    - SDI TESTBED AT COLORADO SPRINGS
    - Ada RELATED WORKING GROUPS, ARTWG, KIT (ETC.)
    - OTHERS AS DIRECTED
- SHOWCASE STATE-OF-THE-ART SOFTWARE SUPPORT ENVIRONMENT
- ASSIST NASA IN REPRESENTATION ON BOARDS AND COMMITTEES
- UHCL CONTACT: DR. CHARLES McKAY

PROOF-OF-CONCEPT PROTOTYPE OF THE CLEAR LAKE MODEL FOR Ada RUN TIME SUPPORT ENVIRONMENTS

- ESTABLISH DISTRIBUTED Ada TESTBED UTILIZING THE EXISTING DMS TESTBED RESOURCES
  - DG MV8000 AS HOST
  - THREE DG MV2000 WORKSTATIONS AS "TARGETS"
  - OFF THE SHELF ETHERNET AS NETWORK
  - ROLM/DG Ada DEVELOPMENT SYSTEM
  - ROLM ARTS REAL TIME OPERATING SYSTEM
  - ROLM Ada RUN TIME ENVIRONMENT
  - ROLM CROSS COM TEST MANAGEMENT SOFTWARE
- ANALYZE AND PROTOTYPE THE EXTENDED Ada RUN TIME ENVIRONMENT IN THE AREAS OF:
  - COMMAND LANGUAGE INTERFACE
  - NON FUNCTIONAL REQUIREMENTS
  - SYMBOLIC DEBUGGING
  - CONSULTING AND DIAGNOSTIC DEMONS
  - SIM BUFFER FUNCTIONALITY
  - HOST/TARGET FUNCTIONALY REQUIREMENTS SPECIFIC TO THE DISTRIBUTED COMPUTER PROBLEM
- UHCL CONTACT: DR. CHARLES McKAY

CONVERSION OF ART TO Ada

- INVESTIGATE THE PROBLEMS TO BE OVERCOME IN CONVERTING AN EXISTING STATE-OF-THE-ART EXPERT SYSTEM DEVELOPMENT TOOL TO Ada
- PHASE 1: DESIGN THE CONVERSION TO Ada OF THE "AUTOMATED REASONING TOOL" OR ART (A PRODUCT OF THE INFERENCE CORPORATION)
  - CURRENTLY WRITTEN IN INFERENCE LISP
  - ANALYSIS AND PRELIMINARY DESIGN
  - INITIAL PROTOTYPING
  - MANAGEMENT PLAN
- PHASE 2: DEVELOP AND DELIVER ART IN Ada
- UHCL CONTACT: DR. CHARLES McKAY

BENCHMARKING Ada ON EMBEDDED COMPUTER SYSTEMS

- ASSESS THE QUALITY AND MATURITY OF Ada DEVELOPMENT SYSTEMS AND Ada RUN-TIME ENVIRONMENTS
- DEVELOP AND COLLECT A SET OF BENCHMARK PROGRAMS
- WITH ASSISTANCE FROM CSDL PERSONNEL RUN THE BENCHMARKS AND REPORT ON FINDINGS
  - MICROVAX / ELN
  - DG MV8000 (ROLM HAWK EMULATION) / ROLM ARTS
  - AN/YUK 44 WITH SOFTECH Ada RUNTIME ENVIRONMENT

Ada RELATED INFORMATION DATABASE

- TO BE AN ON-LINE UTILITY AVAILABLE TO ALL IN THE Ada/SOFTWARE ENGINEERING COMMUNITY
- ORACLE BASED
- LOCATED EITHER ON A CIN OR UHCL MAINFRAME
- HIGHLY USER FRIENDLY - EXTENSIVE MENUS AND REPORT FORMATS AVAILABLE
- EDUCATION AND TRAINING RESOURCES
- Ada RELATED BOOKS AND PAPERS
- DESCRIPTIONS AND INFORMATION ON CURRENT PROJECTS WHERE Ada IS BEING USED
- DESCRIPTION OF REUSABLE Ada COMPONENTS
- UHCL CONTACT: DR. ANTHONY LEKKOS

DEVELOPMENT OF DMS TESTBED USERS MANUAL

- CRITIQUE AND ANALYSIS OF EXISTING TESTBED USERS MANUAL
  - Ada PERSPECTIVE
  - SOFTWARE ENGINEERING PERSPECTIVE
- PROVIDE OVER-ALL ANALYSIS
- PROVIDE SPECIFIC REVISIONS FOR SECTIONS ON:
  - NETWORK COMMUNICATIONS
  - DATA ACQUISITION AND DISTRIBUTION
  - FILE MANAGEMENT
  - DATABASE MANAGEMENT
- UHCL CONTACT: DR. CHARLES McKAY

Ada BETA TEST TEAM SITE SUPPORT

- OBJECTIVE: TO UNDERSTAND AND, WHERE APPROPRIATE, INFLUENCE THE IMPROTANT EVOLVING Ada STANDARDS, GUIDELINES, AND POLICIES SO THAT THEY WILL REFLECT THE BEST INTEREST OF THE SPACE STATION PROGRAM

- INTERFACE WITH FOLLOWING ORGANIZATIONS:
    - Ada BOARD
    - KERNAL INTERFACE TEAM
    - COMMON APSE INTERFACE SET (CAIS) WORKING GROUPS
    - Ada RUNTIME WORKING WORKING GROUP
    - Ada ANSI STANDARDS COMMITTEE
    - ISO Ada WORKING GROUP #9

- UHCL CONTACT: DR. CHARLES McKAY

ESTABLISHED TASKS - ARTIFICIAL INTELLIGENCE

- COMMUNICATION AND TRACKING EXPERT SYSTEM STUDY, PHASE 1
- ANALYSZE THE USE OF EXPERT SYSTEM TECHNIQUES FOR CONFIGURATION AND RECONFIGURATION OF THE SPACE STATION COMMUNICATIONS AND TRACKING SYSTEM
  - ANTENNA TRACKING AND POINTING
  - TV CAMERA POINTING
  - MONITORING AND RECORDING INCLUDING OBSCURATION CONSIDERATIONS
  - AUDIO NETWORK STRUCTURE
  - HISTORICAL DATABASE MANAGEMENT
- UHCL CONTACT: DR. T.F. LEIBFRIED

ESTABLISHED TASKS - EDUCATION AND TRAINING

- SOFTWARE ENGINEERING AND Ada TRAINING
- DESIGN AND IMPLEMENT A PROTOTYPE COURSE FOR UPPER LEVEL MANAGERS ON Ada AND SOFTWARE ENGINEERING (APPX. A HALF DAY)
- ESTABLISH AND MAINTAIN A DATABASE OF SOFTWARE ENGINEERING AND Ada TRAINING AVAILABLE NATIONALLY
- ESTABLISH AND MAINTAIN A DATABASE OF EXPERT RESOURCES IN Ada AND SOFTWARE ENGINEERING AVAILABLE NATIONALLY
- PERFORM AND PUBLISH A MARKET SURVEY OF JSC/CONTRACTOR TRAINING NEEDS IN SOFTWARE ENGINEERING
- DEVELOP AND MAINTAIN A COST-BENEFIT ANALYSIS FOR THE TRAINING OPTIONS IN THE DATABASE
- DEVELOP AND MAINTAIN A STRATEGIC PLAN FOR DELIVERING SOFTWARE ENGINEERING TRAINING TO THE JSC COMMUNITY
- DEVELOP NEW COURSE OFFERINGS AS DIRECTED
- UHCL CONTACT: DR. GLENN FREEDMAN

ESTABLISHED TASKS - INFORMATION MANAGEMENT

- SPACE MARKET MODEL DEVELOPMENT PROGRAM
- CLEAR LAKE AREA COMPUTER CAPABILITY SURVEY

SPACE MARKET MODEL DEVELOPMENT PROGRAM

- PROVIDE AN INFORMATION SERVICE WHICH MONITORS THE EMERGING SPACE MARKET
- PRELIMINARY PRODUCT DEFINITION
- ECONOMIC MODEL OF THE CORE SPACE MARKETS
- FRAMEWORK FOR COLLECTING DATA ON FUTURE SPACE MARKETS
- INTERFACING WITH OTHER INFORMATION SERVICES HAVING DATA ON SPACE MARKETS
- UHCL CONTACT: DR. PETER BISHOP

CLEAR LAKE AREA COMPUTER CAPABILITY SURVEY

- CONDUCT CENSUS OF CLEAR LAKE AREA COMPUTER CAPABILITIES
- AREAS COVERED:
  - HARDWARE
  - SOFTWARE
  - PERSONNEL
  - INSTITUTIONAL AND RESEARCH DIMENSIONS
- ESTABLISH AND MAINTAIN A DATABASE OF THE FINDINGS
- UHCL CONTACT: DR. GLEN HOUSTON

PLANNED TASKS

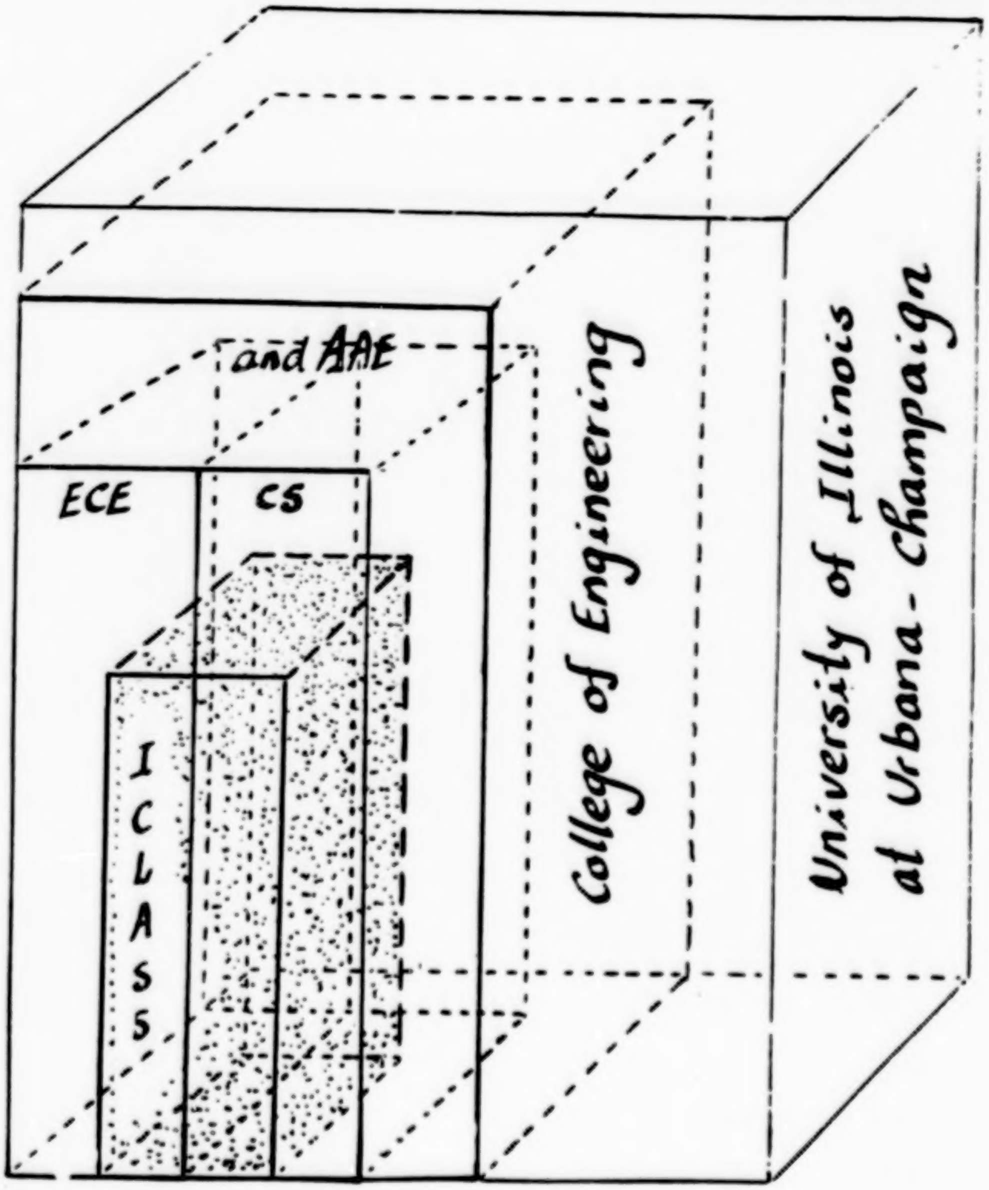
- ARTIFICIAL INTELLIGENCE:
  - COMPUTER GRAPHICS TESTBED TO SIMULATE AND TEST VISION SYSTEMS FOR SPACE APPLICATIONS - UHCL CONTACT: DR. TERRY FEAGIN
  - ROBOT PATH PLANNING - UHCL CONTACT: DR. TERRY FEAGIN
  - APPLICATION OF FUZZY SETS AND RELATED THEORY TO FAILURE DETECTION AND CONTROL IN SPACE SYSTEMS - UHCL CONTACT: DR. GLENN HOUSTON
- INFORMATION MANAGEMENT
  - COMMERCIAL PAYLOADS DATABASE - UHCL CONTACT: DR. PETER BISHOP
- MATHEMATICAL AND STATISTICAL ANALYSIS
  - SPACE STATION MOMENTUM MANAGEMENT AND ATTITUDE CONTROL - UHCL CONTACT: DR. GLEN HOUSTON
- COMPUTER SYSTEMS AND SOFTWARE ENGINEERING
  - SYMBOLIC INFORMATION MANAGEMENT IN Ada TESTING
  - AUTOMATED SOFTWARE VERIFICATION TOOLS
  - FURTHER ANALYSIS OF EXPERT SYSTEMS IN Ada
  - SECURITY ISSUES FOR SPACEBORNE COMPUTER NETWORKS
  - UHCL CONTACT: DR. CHARLES McKAY

SUMMARY AND DISCUSSION

- THE RICIS COOPERATIVE AGREEMENT HAS PROVEN TO BE AN EFFECTIVE METHOD OF ESTABLISHING RESEARCH ACTIVITY AT JSC
- THE NEED FOR RESEARCH IN COMPUTING AND INFORMATION SYSTEMS WILL REMAIN HIGH
  - SOFTWARE LABOR ABSORBING EVER INCREASING PERCENTAGE OF PROJECT BUDGETS
  - DRAMATICALLY INCREASING CAPABILITY OF COMPUTING HARDWARE
- ONE BIG QUESTION: WHAT WILL BE THE EFFECT OF MOVING FORWARD IN THE SHUTTLE AND STATION PROGRAMS
  - AS SHUTTLE RETURNS TO OPERATIONAL STATUS AND STATION MOVES INTO DEVELOPMENT
    - PRESSURE SHIFTS TO GETTING JOB DONE
    - CONCERNS ABOUT USING LATEST TECHNOLOGY WANE
- TO ACHIEVE AND MAINTAIN EXCELLENCE IN COMPUTING TECHNOLOGY NASA'S COMMITMENT TO RESEARCH MUST REMAIN STRONG

ICLASS

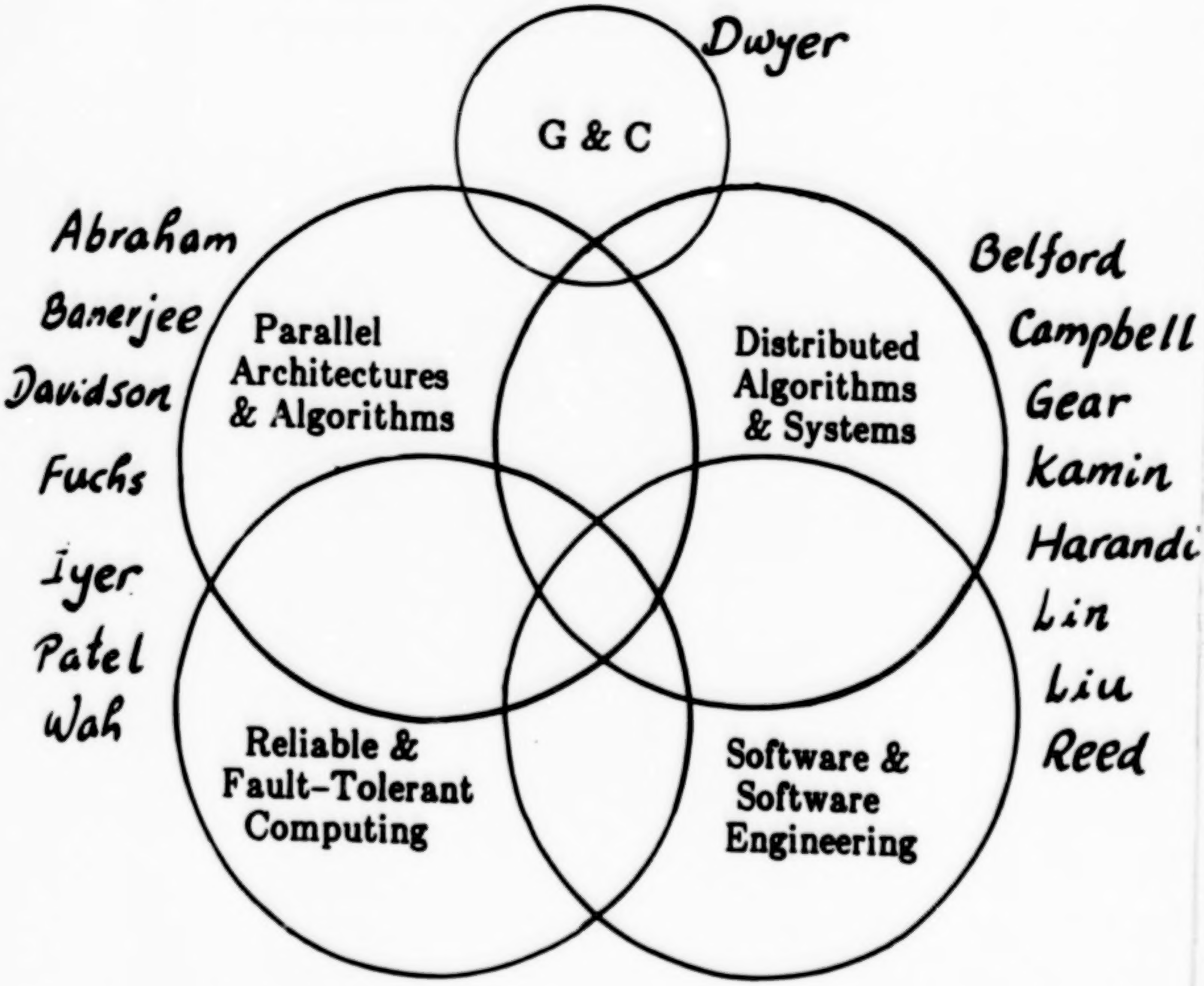
Illinois Computing Laboratory of Aerospace Systems and Software



and AAE
ECE
CS
ICLASS
College of Engineering
University of Illinois
at Urbana-Champaign

Objectives in education and research

ICLASS RESEARCH ACTIVITIES



Dwyer

G & C

Abraham
Banerjee
Davidson
Fuchs
Iyer
Patel
Wah

Parallel Architectures & Algorithms

Distributed Algorithms & Systems

Belford
Campbell
Gear
Kamin
Harandi
Lin
Liu
Reed

Reliable & Fault-Tolerant Computing

Software & Software Engineering

DISTRIBUTED SYSTEMS

On
Dist.
DBMS

- Reliable, Distributed Data Base Management Systems (*Belford*)
- Distributed File System for A Network of Workstations (*Iyer*)

on
Dist.
OS

- Adaptive Operating System for Message Passing Machines (*Reed*)
- Process Migration (*Liu*)
- Distributed, Macro Data Flow Architecture (*Liu*)

On
Comm.
Subnet

- Integrated Communication Networks (*Reed*)

SOFTWARES AND SOFTWARE ENGINEERING

- A Type System for A Functional Programming Language (*Campbell*)
- Environments for Parallel Computation (*Gear*)
- A Program Development System (*Kamin*)
- Abstract, Executable Specification (*Kamin*)
- A Decentralized Debugging Service (*Lin*)
- Software Engineering Data Base and Project Management Support Systems (*Belford, Harandi, Liu*)

- Software design methods to support software reuse

- Reuse software libraries

  ISADORA – numerical software

  Clemma – part of SAGA general library

- Interconnection language

  Polylith – for interconnection of different modules on dist. systems

- Open System architecture

- Object-base software engineering database supporting version, change monitoring and notification.

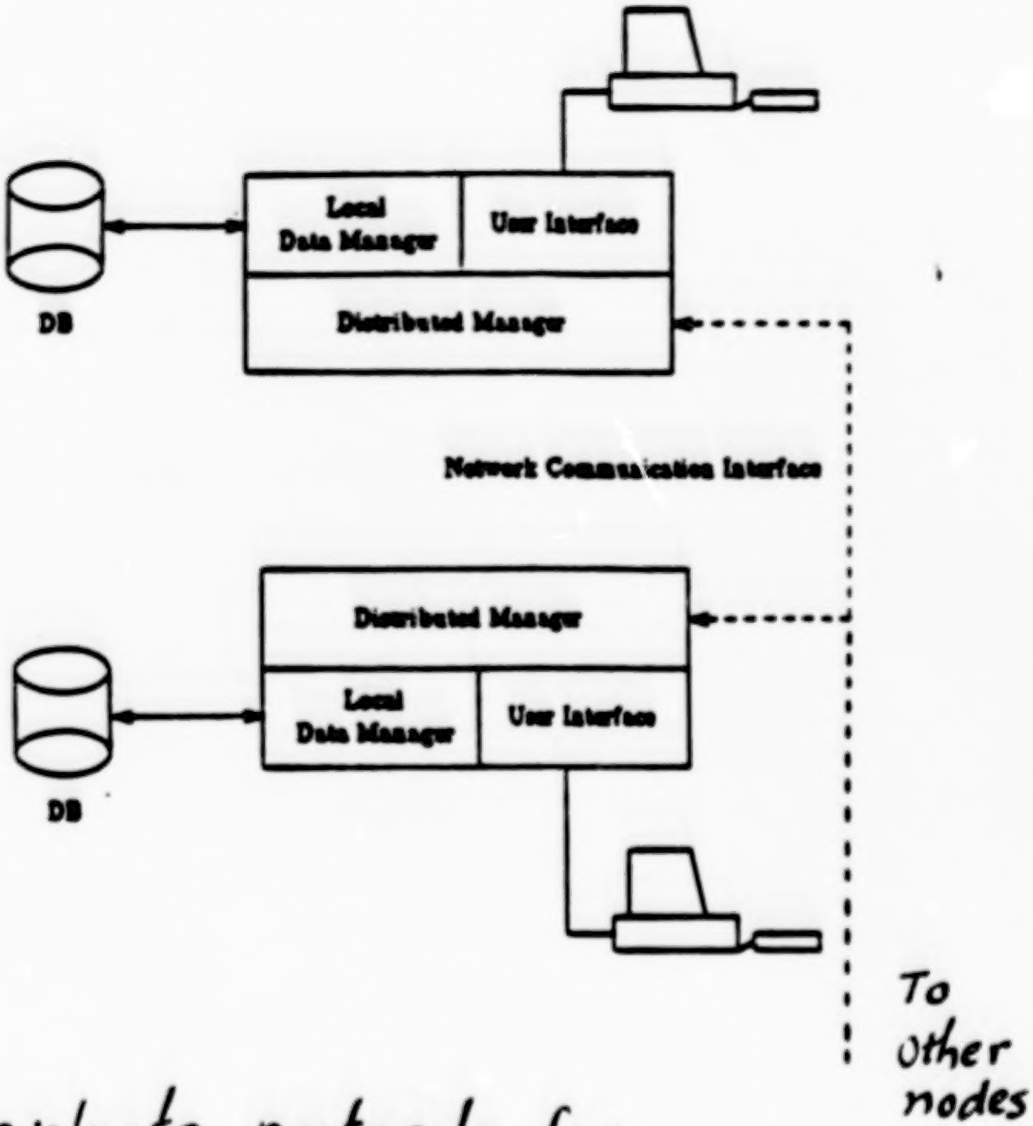
HIGH PERFORMANCE AND HIGHLY AVAILABLE DISTRIBUTED SYSTEMS

- Distributed Task Assignments
- Load Balancing Algorithms
- Dynamic Control Structures
- Reconfiguration Schemes
- Process Migration
- Objected–Oriented, Data Flow Architecture

Three Implementation Projects
to build
DISTRIBUTED SYSTEMS

- *DROLL*, A Distributed Testbed for DDBMS Algorithms and Protocols.
- *PICASSO*, A Adaptive Distributed Operating System
- *MENTAT*, A Object–Oriented, Macro Data Flow System

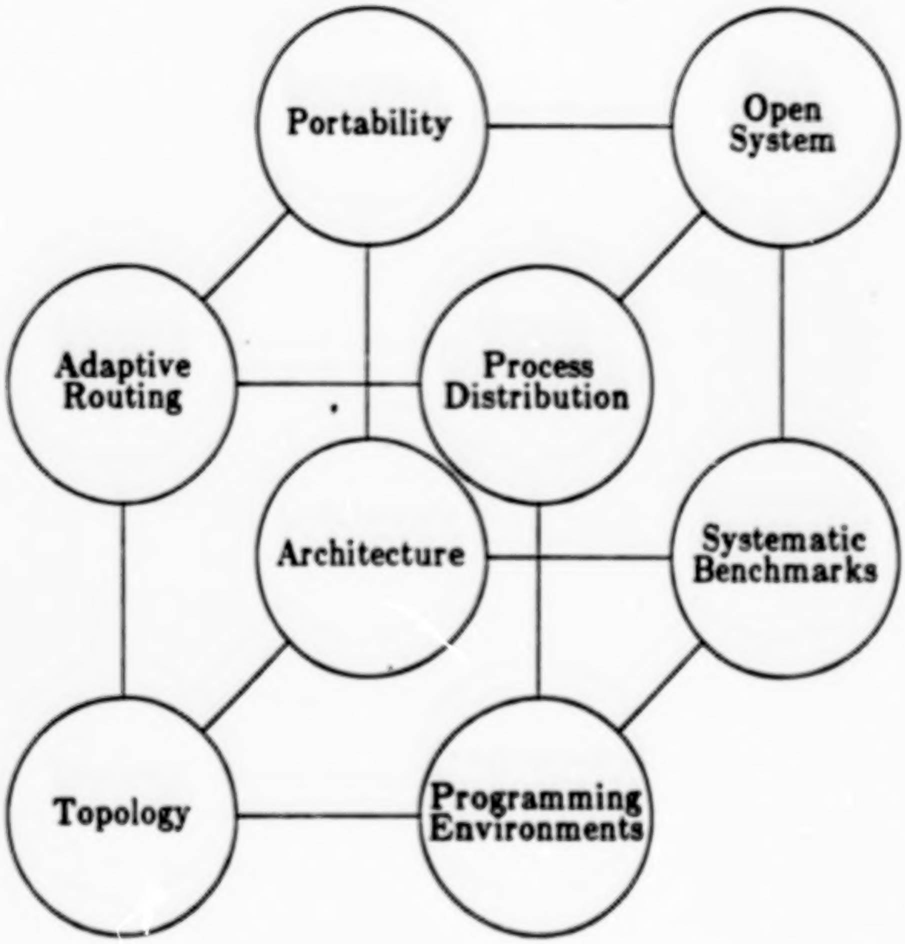
DTROLL, A Test Bed


Local Data Manager
User Interface
Distributed Manager
DB
Network Communication Interface
Distributed Manager
Local Data Manager
User Interface
DB
To other nodes

To evaluate protocols for

- Concurrency Control
- Failure detection and recovery

The Cornerstones of
Picasso


Portability
Open System
Adaptive Routing
Process Distribution
Architecture
Systematic Benchmarks
Topology
Programming Environments

Picasso Will Provide

- A frame–work for testing and evaluating message routing & load sharing methods

- An **open**, **layered**, and **expandable** basis for operating system design and experimentation

- A **portable, efficient** environment for parallel algorithm development

Only by advancing the state of the art in programming environments for message passing computers can we realize the promise of effective large–scale parallelism.

Using adaptive routing, dynamic load sharing and a consistent system representation, **Picasso** will engender solutions which harness parallelism while maintaining clarity and reliability.

MENTAT

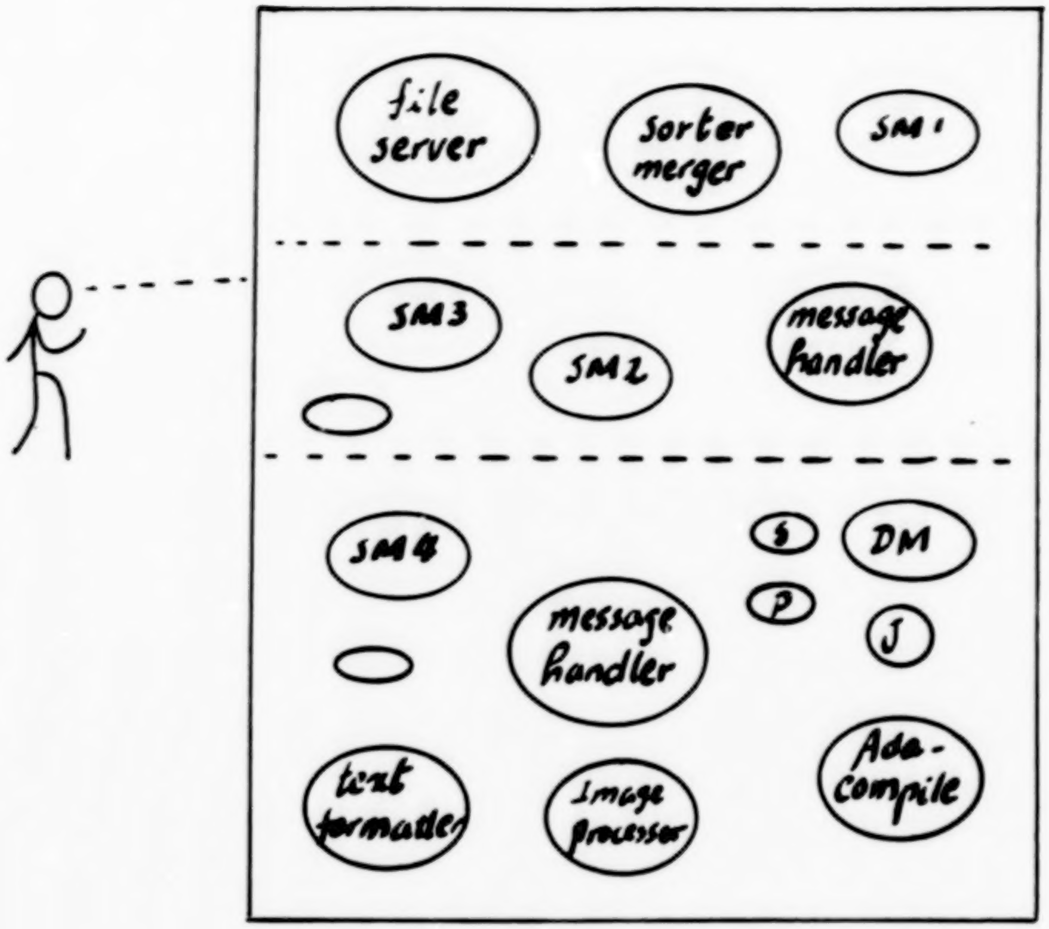
An Objected–Oriented, Data Flow Machine

OBJECTIVES

- To provide an environment conducive to distributed software development and effectively supports software reuse

- To develop a high performance distributed system in which the degree of parallelism is maximized

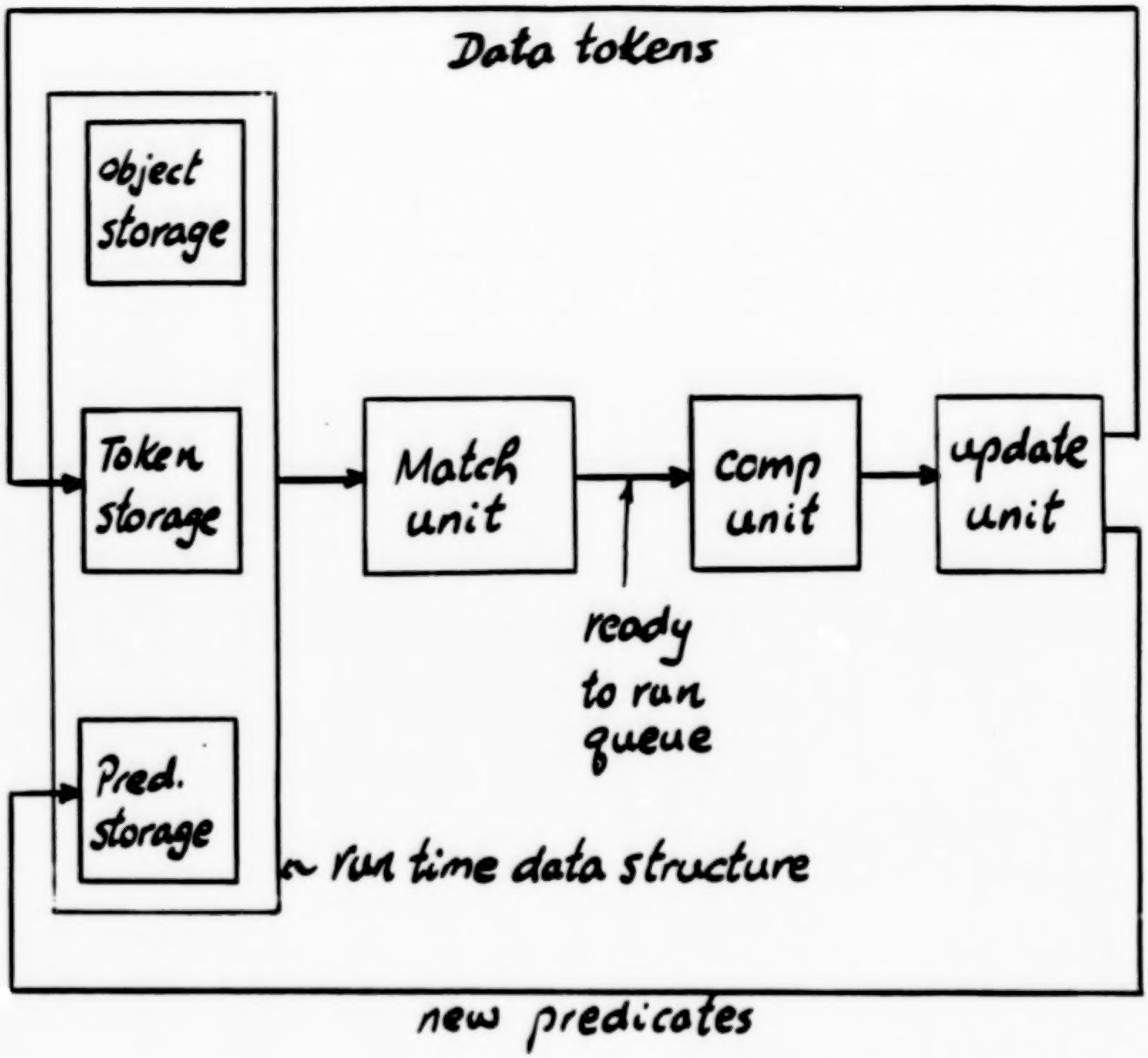
Distributed Macro Dataflow Machine


file server
Sorter merger
SM1
SM3
SM2
message handler
SM4
S
DM
P
J
message handler
Ada-compile
text formatter
Image processor

- Components of MENTAT:
  - Actor - Objects
  - Token or messages
- Programming Languages:

  C++ and Ada

A Macro Data Flow Machine


Data tokens
object storage
Token storage
Pred. storage
run time data structure
Match unit
comp unit
update unit
ready to run queue
new predicates

Run Time Support in MENTAT

Summary of ICLASS Activities

Systems and **Software Engineering**

An adaptive DOS

A Macro data flow architecture

Process migration

Distributed file systems for networks of workstations

Reliable, distributed DBMS

Environments for Pa. Pro.

A type system for a FP

Abstract, executable specification

A program development system

A decentralized debugging service

Software engineering data base and Project management support systems

OVERVIEW OF ICLASS RESEARCH: RELIABLE AND PARALLEL COMPUTING

R. Iyer

Computer System Group
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

ICLASS ACTIVITIES IN RELIABLE AND PARALLEL COMPUTING

- **6 Faculty, 10 students**
- **Research Areas**
  - Reliable Computing
  - Parallel Processing
- **Interaction with NASA Centers**
  - Student/Faculty visits to NASA facilities
  - Joint Research with NASA-Ames, JPL
  - Interactions with NASA Langley
- **Other Activities**

RELIABLE AND FAULT TOLERANT COMPUTING

- **Automatic Recognition of Failure Symptoms**
  - ☐ Real time prediction of major system problems based on low-level error/recovery symptoms
  - ☐ Prediction based on learning from observed behavior

- **Structure Based Error Detection**
  - ☐ Detection of major software problems
  - ☐ Detection of hardware failures affecting software integrity

- **Algorithm-Based Fault Tolerance**
  - ☐ Takes advantage of unique features of the algorithm
  - ☐ Cost effective for specific application e.g. signal processing, Matrix Manipulations

RELIABLE AND FAULT TOLERANT COMPUTING

• Fault Latency: Experimental Study

- ☐ Simulation based study using real workload data
- ☐ Accurate evaluation of fault latency distributions
- ☐ Latency distributions valuable for accurate reliability prediction and estimation of multiple failures

PARALLEL PROCESSING - APPLICATIONS

- **Image Processing**
    - ☐ Low level operations e.g. convolution, filtering
    - ☐ High level operations e.g., connected components
    - ☐ Unique feature is handling both in real time
- **Design Automation**
    - ☐ Standard Cell Placement
    - ☐ Wire Routing
    - ☐ Unique feature is use of hypercube to give large speedup instead of special purpose CAD accelerators
- **AI Algorithms and Software**
    - ☐ Development of heuristic rules to guide parallel run-time search for LISP and PROLOG programs
    - ☐ Investigation of architectures to support parallel search techniques

PARALLEL PROCESSING

- **Architecture**
  - ☐ Reduction in communication overhead
  - ☐ Efficient use of local memory
  - ☐ Study of memory referencing characteristics for individual locations
- **Measurement-based Evaluation**
  - ☐ Concurrency Measurements
    - Hardware Monitoring
    - Software Instrumentation
  - ☐ Relating concurrent program behavior to system resource usage

AUTOMATIC RECOGNITION OF FAILURE SYMPTOMS

R. K. Iyer, L. T. Young and V. Sridhar

Computer Systems Group
Cooridinated Science Laboratory
University of Illinois at Urbana-Champaign

OBJECTIVES

- Recognize onset of persistent system problems based on observed relationships between low level recoverable errors

- Differentiate between transients and intermittents

- Quantify the strength of the captured relationships

BACKGROUND/MOTIVATION

- Early recognition of symptoms of persistent errors usually not possible

- Causes usually complex—may affect different parts of a system

- Detection commonly occurs as many isolated incidents

- Incorrect diagnosis leads to improper recovery management --> affects system integrity

- To be effective, the system must examine seemingly unrelated error events for possible relationships

SYMPTOM RECOGNITION

1. Clustering of identical errors

2. Identification of sudden rise in failure rate— **GROUPING**

3. Identification of **EVENTS**—"Similar" groups

4. Identification of **SUPER EVENTS**—"related" events

5. Quantification of the strength of inter-record relationships

VALIDATION OF FAILURE SYMPTOMS

- **Probabilistic Validation**

$$P(A_1)*P(A_2)*...*P(A_n) < P(A_1, A_2, \cdots, A_n)$$

$$\textbf{Strength (S)} = \frac{P(A_1)*P(A_2)*...*P(A_n)}{P(A_1, A_2, \cdots, A_n)}$$

- **Group Level Validation**
  - ☐ Complete Analysis
  - ☐ Subset-Based Analysis

- **Event-Level Validation**
  - ☐ Inter-Group Analysis

- **Super Events**
  - ☐ Inter-Event Analysis

SUMMARY

1. Demonstrated a technique for fault isolation
2. Allows resolution between independent and dependent causes, transients and intermittents
3. Estimates strength of relationship among dependent failures (useful in prediction)
4. Can evaluate effectiveness of new diagnostic techniques

FUTURE RESEARCH

- Current results show that real-time diagnosis is feasible
- Develop strategies for on-line diagnosis and reconfiguration for complex multi-computer systems
- Perform hardware and software error injection experiments on network of workstations
- Extension to symbolic architectures
- Investigate feasibility of on-line software "repair" routines

What is CASIS?

- CASIS is Stanford's Center for Aeronautics and Space Information Sciences

- CASIS was founded in March, 1983 with the support of a block grant from NASA's Office of Aeronautics and Space Technology

- CASIS was founded as a multi-disciplinary, cross-departmental research structure which would provide faculty and students with an opportunity to participate in a variety of space-related projects relating to activities of the U.S. civil space program

- CASIS includes 13 Stanford faculty, 9 professional research staff, 31 graduate students, and 6 undergraduates

- The CASIS administrative offices, the central computer facilities, and the CASIS Experiment Laboratory are located in Durand Building. Participating projects are located in various CSD and EE buildings on campus

CASIS Governance

- CASIS is governed under a charter approved by the Stanford School of Engineering and the Provost of the University

- The overall direction of the center is provided by the Director, Prof. Peter M. Banks.

- Day to day operations are under the direction of the Associate Director, Dr. Michael J. Wiskerchen.

- Policy decisions for CASIS are reviewed by the CASIS project investigators

- Progress of research projects and associated budgets are monitored by the Director in collaboration with individual research project leaders

- The practical responsibility for research work lies in the hands of the research leaders

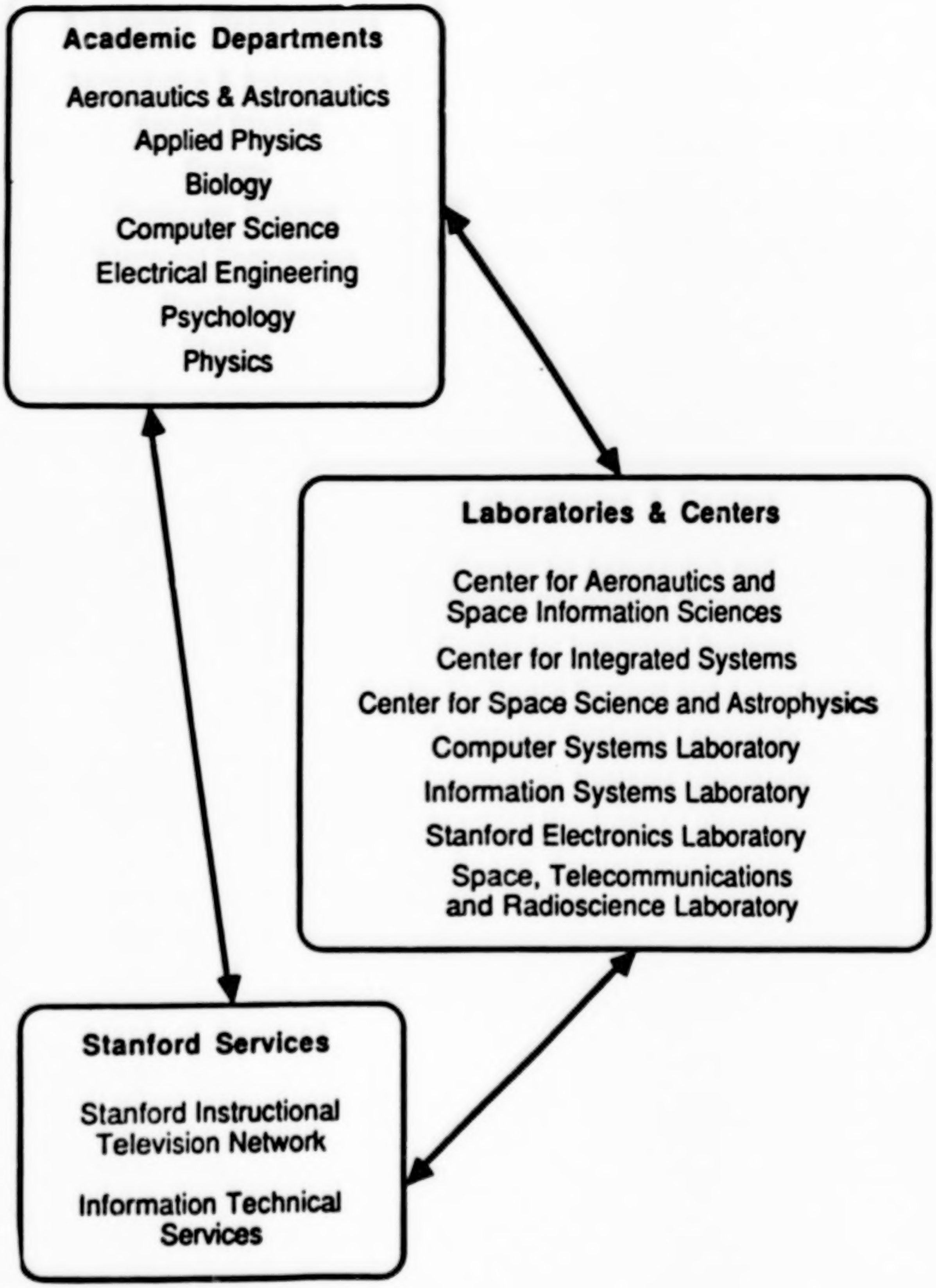
- The Director reports to the Associate Dean for Research in the School of Engineering

CASIS

- Center for Aeronautics and Space Information Sciences
- NASA OAST HQ block grant in information sciences
- Advanced information sciences technology research
- Core technology for "TELESCIENCE" applications research
- Advanced technology for Shuttle and Space Station applications

ORIGINAL PAGE IS
OF POOR QUALITY

STANFORD RESOURCES



Academic Departments

- Aeronautics & Astronautics
- Applied Physics
- Biology
- Computer Science
- Electrical Engineering
- Psychology
- Physics

Laboratories & Centers

- Center for Aeronautics and Space Information Sciences
- Center for Integrated Systems
- Center for Space Science and Astrophysics
- Computer Systems Laboratory
- Information Systems Laboratory
- Stanford Electronics Laboratory
- Space, Telecommunications and Radioscience Laboratory

Stanford Services

- Stanford Instructional Television Network
- Information Technical Services

Center for Aeronautics and Space Information Sciences
CASIS

- Concurrency Studies
  - Researchers: M. Flynn, S. Owicki, S. Lunstrom - 5 students
  - Performance of Highly-Concurrent Systems
  - Models of Parallel Programs
  - Highly Reliable Concurrent Programs
- Network Graphics and User Interface Architecture
  - Researchers: K. Lantz - 4 students
  - Virtual Graphics Terminal Service
  - User interface architecture: Terminal management - Command specification and response handling - application specific interaction - multi-application management

CURRENT CASIS RESEARCH

- Studies in Computer Concurrency-- Prof. M. J. Flynn
- High Speed Local Area Networks-- Prof. F. A. Tobagi
- High Performance VLSI Signal Processors-- Prof. A.M. Peterson
- Parallel Computing Arrays for Signal Processing-- Prof. T. Kailath
- Satellite Communications for Space Science Data Systems-- Prof. B.B. Lusignan
- Network Graphics and User Interface Architecture-- Prof. K.A. Lantz
- Large Volume Database Management-- Prof. G. Wiederhold
- Rapid Image Retrieval for Remote Sensing Data-- Prof. P.M. Banks
- Advanced Video Systems and Robotic Vision Systems for Scientific Analysis-- Dr. M.J. Wiskerchen
- Highly Concurrent Systems-- Prof. S.F. Lunstrom
- Reliable Concurrent Programs-- Prof. S.S. Owicki

M. J. Wiskerchen — STARLAB/SEL

Center for Aeronautics and Space Information Sciences

CASIS

- High Speed Local Area Networks
    - Researchers: F. Tobagi, M. Mahric - 4 students
    - Topological Design of Fiber Optics LANs
    - Study of Media Access Protocols
    - Design and Implementation of the Network Interface
    - EXPRESSNET implementation
- Telecommunications Research
    - Researchers: B. Lusignan - 5 students
    - Efficiency modeling of satellite digital transmission
    - Study of new modulation techniques
    - C-MOS integrated circuit design for radio-channel equipment
    - Efficiency models for fiber-optic digital networks
    - Optimization modeling for digital communications components
    - Digital Telecommunications switch development

M. J. Wiskerchen — STARLAB/SEL

Center for Aeronautics and Space Information Sciences

CASIS

- Parallel Computing Arrays for Signal Processing
  - Researchers: T. Kailath - 2 students
  - Parallel Algorithms and Architectures for Kalman Filtering
  - Schur vs. Levinson Algorithms for Stationary and Nonstationary Processes in Data Communications
- Advanced Concepts for Remote Space Science Operations
  - Researchers: P. Banks, P. Williamson, R. Clauer, M. Wiskerchen - 6 students
  - Telescience Research for Shuttle and Space Station Systems
  - Distributed Computer Graphics Networks

Center for Aeronautics and Space Information Sciences
CASIS

- High Performance Signal Processors
    - Researchers: A. Peterson, I Linscott - 3 students
    - PROLOG language as a processor description tool and VLSI CAD systems
    - High performance architectures with relational processing
    - Real time parallel processing - Scheduling Theory approach
    - Optimization of pipelined processor using linear programming
- DataBase Management Architectures
    - Researchers: G. Wiederhold, P. Rathman, H. Waquar - 4 students
    - Database design and operation methods and algorithms
    - Optical storage technology

New CASIS Research Topics

1. Color Stereoscopic Video Systems-- Prof. P.M. Banks, EE
2. Speech Recognition and Analysis-- Prof. R. M. Gray, EE
3. VLSI Tactile Sensors-- Dr. P.W. Barth, CIS
4. ADA Development for Space Systems-- Prof. D. C. Luckham, EE
5. Distributed Network Architectures for A.I. Work Stations --Dr. T.C. Rindfleish, CSD
6. Human Factors for Space Experiment Operations-- Dr. M. J. Wiskerchen, EE

TELESCIENCE

Perform "Science in Space" like in a ground-based laboratory throughout the entire life cycle; from instrument development through operations to the reduction of data to new knowledge and its publication

Definition:

The direct, iterative and distributed interaction of users with their instruments, data bases, specimens and data handling facilities, especially where remote operations are essential

The distributed interaction is meant to include ALL members of a user team, in space and on the ground, and may involve either manned or unmanned operations. It is the general desire of the user community to conduct their operations from their home institution by on-line computer networking.

TELESCIENCE

Can be divided into three portions, centered on pre-flight, flight, and post-flight:

1. Teledesign

The ability to send drawings, documents and specifications, to perform interactive design with remote facilities, and to conduct interface and other tests of instruments by remote computer access.

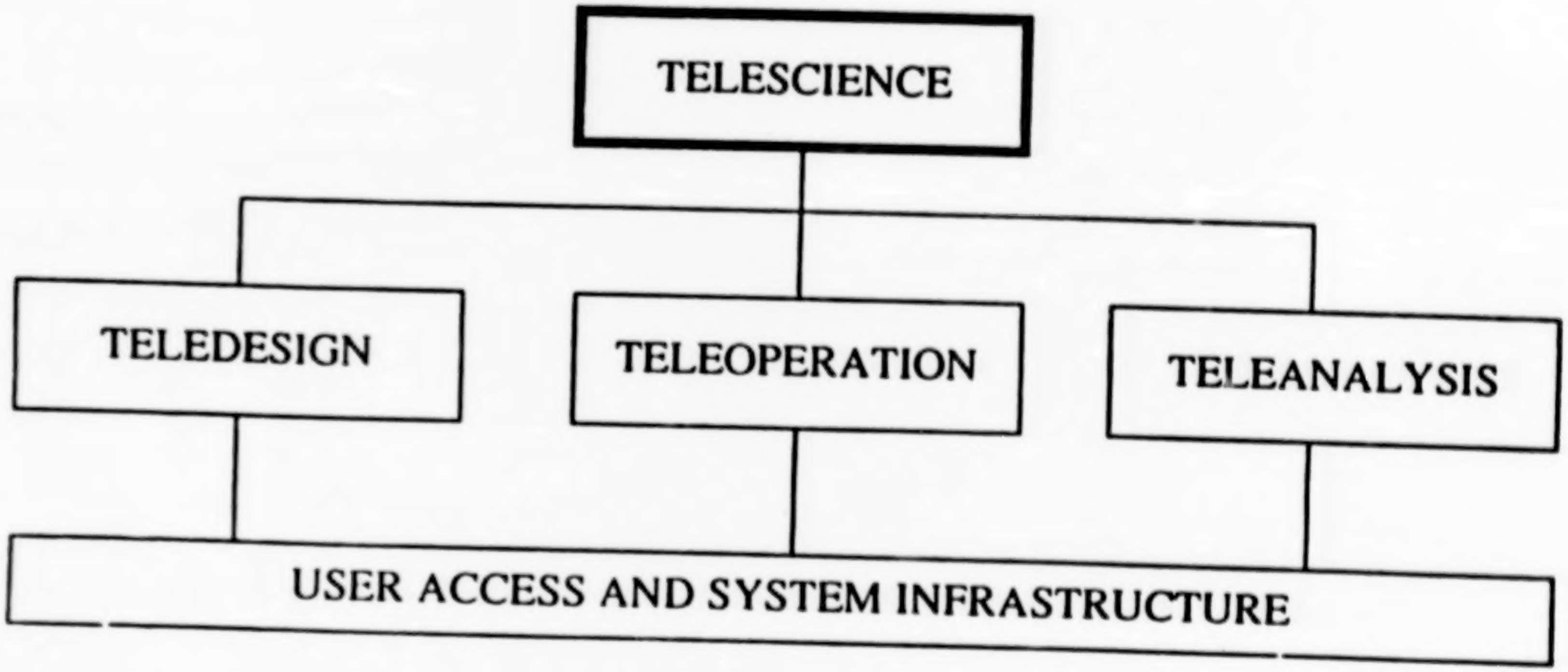
2. Teleoperations

The ability to conduct remote operations by making rapid adjustments to instrumental parameters and experiment procedures in order to obtain optimum performance.

3. Teleanalysis

The ability to access and merge data from distant sources and to perform analyses and studies on computers that may be located at other institutions.

TELESCIENCE CONCEPT



TELESCIENCE

TELEDESIGN

TELEOPERATION

TELEANALYSIS

USER ACCESS AND SYSTEM INFRASTRUCTURE

Stanford University Network for Space Telescience Applications Research

SUNSTAR PROGRAM

M. J. Wiskerchen — STARLAB/SEL

VALUE OF TELESCIENCE

- ENHANCES PRODUCTIVITY OF REMOTE OPERATIONS
  - Gives better use of local and remote human resources
  - Reduces mistakes
  - Provides a capability to quickly investigate new or unexpected phenomena
- SHARPENS FOCUS OF SPACE RESEARCH OPERATIONS
- PROVIDES COORDINATION OF EXPENSIVE RESOURCES
- PROVIDES DESIGN GOALS FOR SYSTEM PLANNERS
- GIVES FOCUS TO NEW TECHNICAL NEEDS TO SUPPORT SCIENTIFIC RESEARCH OPERATIONS

Technical Challenges for Telescience

1. Restricted bandwidth in telecommuncation channels
    - Need for information compression
    - Dynamic channel selection based on useage models
    - New modes of communication (e.g., electro-optics)
    - Use of object coded definitions for graphics

2. Finite signal propagation time
    - Look-ahead capabilities for experiments and operations
    - Autonomous operations
    - Advanced local simulation of remote system behavior

3. Integrated information acquisition and dissemination
    - Digital data to include images, voice, computer
    - Development of appropriate packet techniques
    - Main link vs. local and remote distribution, capture and processing

4. Spatial diversity of user centers for single experiment sites
    - Control and command models
    - Sharing of current information
    - Segregation of returning information to local addressees
    - Standard interfaces and access protocols

5. Human interactions
    - Integration of voice, image and computer information for experiment control
    - Optimization of cognition for "adaptive" science
    - Balance between automatic sequences and timely direct control
    - The value of local simulated and time-delayed experiment Images

6. Tool development
    - Voice coding and synthesis
    - Voice recognition and computer interaction
    - Simulation 3-d graphics
    - Stereoscopic video
    - High definition video
    - Knowledge-based operations planning
    - A.I. for remote instruments and support equipment
    - A.I. for local site system configuration and operation
    - Hardware support for network graphics
    - High-bandwidth local area network for mixed mode data
    - Query language database management

7. Cost issues

    - System costs must match user needs

8. Reliability

    - Needs will vary according to the telecommunications channels used and the type of project

TELESCIENCE

NEW ENGINEERING PHILOSOPHY

A NEW, UNIFYING CONCEPT BASED ON THE NEED TO IMPROVE THE WAY REMOTE RESEARCH OPERATIONS ARE CONDUCTED IN SPACE AND IN A TERRESTRIAL ENVIRONMENT

WILL PROVIDE A DESIGN GOAL FOR SPACE STATION TECHNICAL DEVELOPMENTS

HAS STRONG TIES TO AUTOMATION, ROBOTICS, EXPERT SYSTEMS, COMPUTER VISION, AND OTHER TOPICS RELATED TO KNOWLEDGE-BASED ENGINEERING

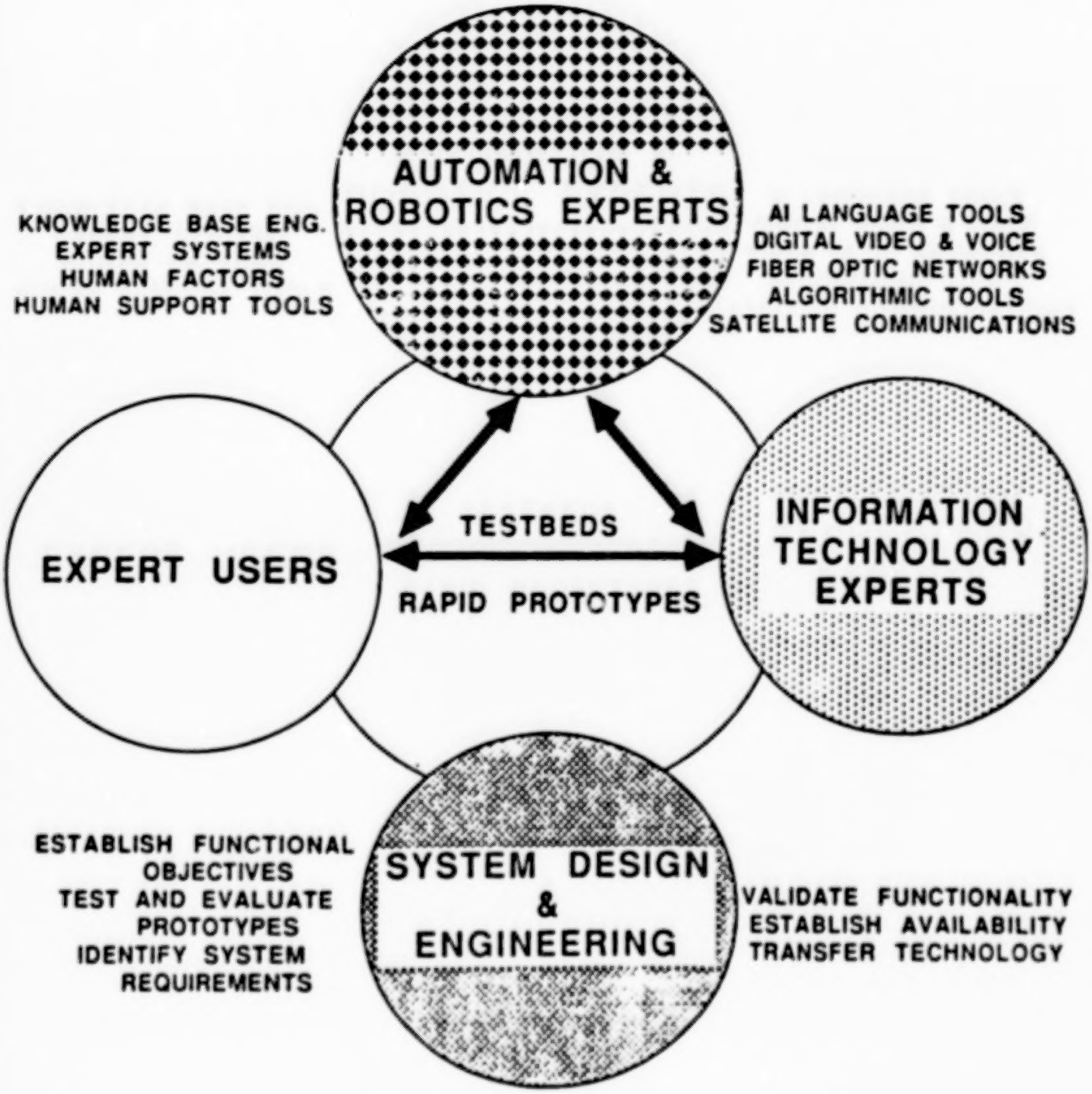
IS AT THE FOREFRONT OF NEW APPLICATIONS INVOLVING HUMAN FACTORS ENGINEERING

PROVIDES A MECHANISM FOR UNIVERSITIES, INDUSTRY AND GOVERNMENT TO WORK TOGETHER ON RESEARCH AND TECHNOLOGY PROGRAMS

PROMOTES THE USE OF RAPID PROTOTYPING CONCEPTS AND TESTBEDS TO DEVELOP FUNCTIONAL SYSTEM DESIGNS

PROVIDES NEW OPPORTUNITIES FOR UNIVERSITY-BASED RESEARCH AND PILOTS

ORIGINAL PAGE IS
OF POOR QUALITY


AUTOMATION & ROBOTICS EXPERTS
KNOWLEDGE BASE ENG.
EXPERT SYSTEMS
HUMAN FACTORS
HUMAN SUPPORT TOOLS
AI LANGUAGE TOOLS
DIGITAL VIDEO & VOICE
FIBER OPTIC NETWORKS
ALGORITHMIC TOOLS
SATELLITE COMMUNICATIONS
EXPERT USERS
TESTBEDS
RAPID PROTOTYPES
INFORMATION TECHNOLOGY EXPERTS
SYSTEM DESIGN & ENGINEERING
ESTABLISH FUNCTIONAL OBJECTIVES
TEST AND EVALUATE PROTOTYPES
IDENTIFY SYSTEM REQUIREMENTS
VALIDATE FUNCTIONALITY
ESTABLISH AVAILABILITY
TRANSFER TECHNOLOGY

UNIVERSITY CONSORTIUM FOR SPACE STATION TELESCIENCE RESEARCH

PURPOSE:

PROVIDE A MEANS FOR THE UNIVERSITY SPACE SCIENCE COMMUNITY TO INVESTIGATE, DEVELOP AND VALIDATE FUNCTIONAL REQUIREMENTS FOR THE DESIGN OF THE SPACE STATION

APPROACH:

DO EXPERIMENTS WITH USERS ON PROTOTYPES OR TEST-BEDS AND EXTRACT DATA FOR EVALUATION OR TRADE-OFF ANALYSIS

REQUIRES A TEAM INCLUDING EXPERIMENT EXPERTS, SYSTEM ENGINEERS, USER ANALYSTS AND PROTOTYPERS

PROCESS:

FRAMEWORK OF THE PROCESS IS AS FOLLOWS:
- OPERATIONS CONCEPT
- USER ANALYSIS
- TASK ANALYSIS
- MAN - MACHINE INTERFACE ANALYSIS
- VALIDATION

THE PROCESS IS NOT LINEAR BUT ITERATIVE AND RECURSIVE AND MUST BE SUPPORTED BY PROTOTYPES WHICH EVOLVE DURING THE PROCESS

Stanford University Network for Space Telescience Applications Research

SUNSTAR PROGRAM

M. J. Wiskerchen — STARLAB/SEL

- RESEARCH IN PAYLOAD DESIGN, INTEGRATION AND OPERATIONS
    - MSFC, GSFC, JSC, KSC, ARC, LeRC cooperative agreements
    - Stanford implements distributed network communications
    - Use Shuttle payloads as testbed for "TELESCIENCE" research

COOPERATIVE AGREEMENT RESEARCH

GODDARD SPACE FLIGHT CENTER

- Remote space science operations facility
- Integrated voice, video, computer work station
- CAD/CAE/CAM work stations
- Human factors research for operations and control systems
- Fiber optics local area networks
- Optical disk mass storage systems
- Parallel processor architectures
- Satellite network architectures
- Expert systems for operations centers
- Expert systems for communications control

MARSHALL SPACE FLIGHT CENTER

- Expert systems for payload timelines
- Local area network architectures
- Payload Operations Control Center studies
- CAD/CAE workstations
- Distributed engineering databases for payloads
- AI workstations

JOHNSON SPACE CENTER

- Fiber optics local area networks
- High performance work stations
- ADA software architectures
- Expert systems for operations and control
- High definition and digital video
- Optical storage systems and databases
- Satellite communications architectures
- Digital speech recognition and synthesis
- Speech and video compression

AMES RESEARCH CENTER

- Robotic vision systems
- Human factors research
- Parallel processor architectures for AI
- Speech recognition and synthesis
- Large computational architectures
- Computational algorithm development

COOPERATIVE AGREEMENT RESEARCH

LEWIS RESEARCH CENTER
- Advanced Communications Technology Satellite -- system architecture study
- Expert systems for space station power system control & operations

KENNEDY SPACE CENTER
- Remote testing and integration research
- Expert systems for integration operations
- Fiber optic local area networks

APPLICATIONS OF CASIS TECHNOLOGY

STANFORD/NASA CENTER COOPERATIVE AGREEMENTS

1. GSFC

   ADVANCED COMMUNICATIONS DESIGNS
   HUMAN FACTORS STUDIES FOR CONTROL CENTERS
   MULTIMEDIA WORKSTATIONS
   PARALLEL COMPUTING ARCHITECTURES
   CAD & CAE GRAPHICS SYSTEMS
   EXPERT SYSTEMS APPLICATIONS TO CONTROL CENTERS
   REMOTE SCIENCE OPERATIONS CENTERS

2. KSC

   AUTOMATION AND ROBOTICS APPLICATIONS TO ORBITER PROCESSING, LAUNCH OPERATIONS, AND PAYLOAD INTEGRATION AND PROCESSING

3. JSC

   MULTIMEDIA OPERATIONS WORKBENCH
   FIBER OPTIC LOCAL AREA NETWORKS
   EXPERT SYSTEM APPLICATIONS
   ADVANCED MISSION CONTROL CENTER DESIGN
   SPACE STATION TELESCIENCE TESTBED PROGRAM

4. MSFC

   ADVANCED CONCEPTS IN PAYLOAD OPERATIONS CONTROL CENTERS AND REMOTE SCIENCE OPERATIONS CENTERS
   EXPERT SYSTEM APPLICATIONS TO PAYLOAD OPERATIONS

5. LeRC

   DESIGN STUDIES FOR THE ACTS PROGRAM

Stanford University Network for Space Telescience Applications Research

SUNSTAR PROGRAM

M. J. Wiskerchen — STARLAB/SEL

- AUTOMATION AND ROBOTICS
  - Timeline engineering using expert systems
  - Vision systems in payload robotics
  - Computer architectures for robotics
  - AI languages
  - Communication networks for robotics
  - Robot systems for payload control
  - Human factors research for man-machine interfacing
  - Expert systems for payload integration

STANFORD
Lockheed

Space Systems Integration and Operations Research Applications (SIORA) Program

The SIORA Program is a joint Stanford/Lockheed initiative to accomplish near term applications of artificial intelligence, robotics, imaging systems, and other research products. Near term applications are to be achieved through rapid (6 to 24 months) development of prototype ground applications. This "rapid prototyping" will provide test beds or pilots for development and demonstration of technological approaches to integration and operation of space systems. Follow-on efforts will be undertaken to develop selected operational applications to benefit current or future space transportation systems and payloads.

BASIC THEMES OF THE SIORA PROGRAM

- AUTOMATION AND ROBOTICS RESEARCH APPLIED TO SPACE SYSTEMS
  - EMPHASIS ON INTEGRATION AND OPERATION
  - GROUND AND FLIGHT SYSTEMS SUPPORT
- APPLICATIONS DRIVEN RESEARCH
  - TASKS BASED ON OPERATIONAL NEED
  - WORK WITH LSOC AND KSC TO DEFINE INITIAL TASKS
- A STANFORD/LOCKHEED TEAMING EFFORT
  - UNIVERSITY/INDUSTRY RESEARCH PARTNERSHIP
  - POTENTIAL MODEL FOR GOAL ORIENTED RESEARCH PROJECTS
- NEAR TERM RAPID PROTOTYPING: 6 TO 24 MONTH TIME FRAMES
- INITIAL SIORA TASKS WITH KSC; OTHER POTENTIAL TASK AREAS INCLUDE A & R RESEARCH APPLICATIONS FOR USAF-SCF, VAFB, AEROSPACE PLANE, DARPA, SDI, ETC.

Institute For Computer Applications In Science and Engineering

ICASE

R. Voigt/LaRC

ICASE

Permanent Staff
R. G. Voigt - Director
M. Y. Hussaini - Chief Scientist

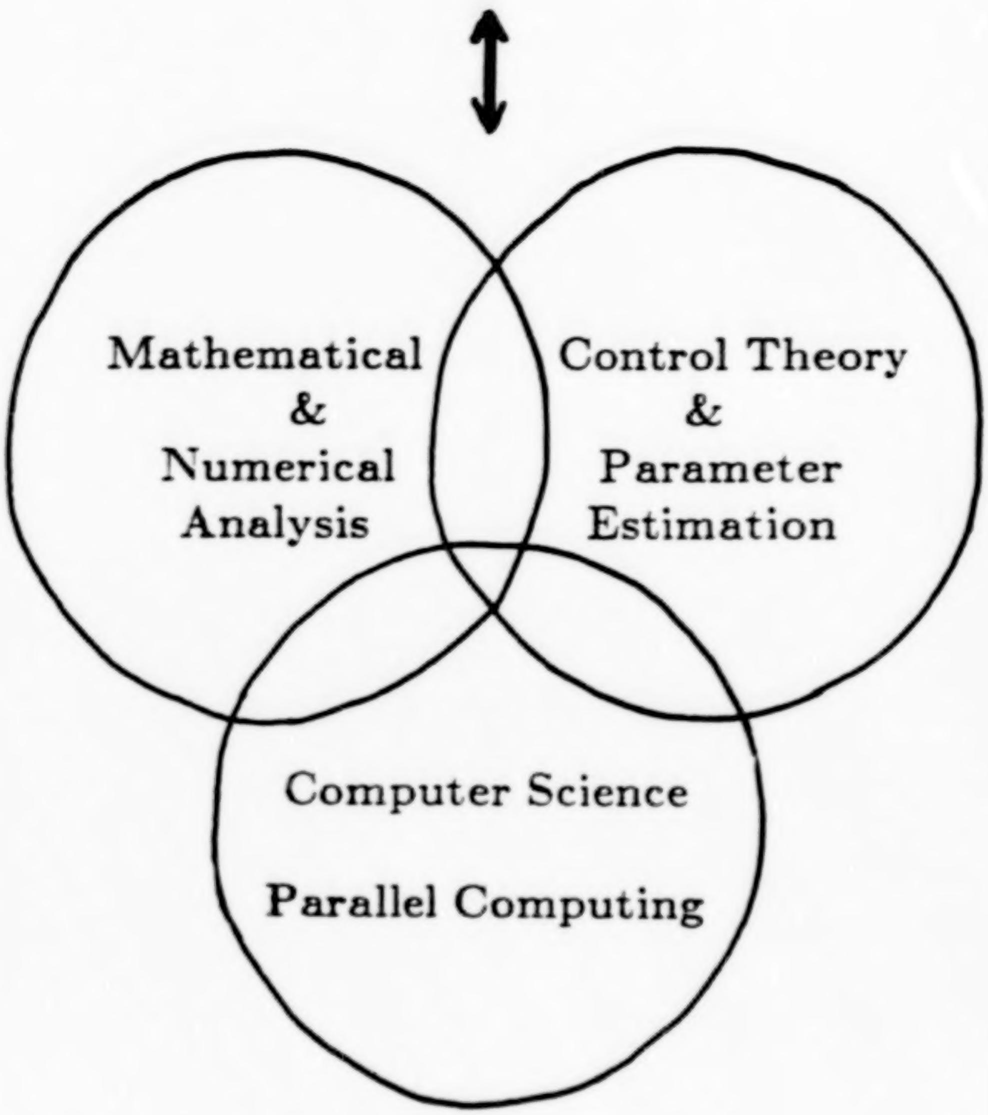
Visiting Staff -- 16-20 Ph.D.'s

Consultants

Summer Program

"Product"
75 Reports
2 - 4 Workshops/Conferences

LaRC Applications



Mathematical
&
Numerical
Analysis

Control Theory
&
Parameter
Estimation

Computer Science

Parallel Computing

VISIBILITY

13 Publications

Second SIAM Conference on Parallel Processing & Scientific Computing

Active participation in variety of other meetings

4 Staff Scientists

10 Consultants

PARALLEL PROCESSING

Systems Issue

Alternative mathematical models

Numerical algorithms

Models of computation

Mapping problem

Dynamic load balancing

Communication & synchronization requirements

Programming environments

NUMERICAL ALGORITHMS

Modify order of computation to

1) Increase parallelism

2) Decrease communication

3) Decrease synchronization

Numerical algorithms used for definiteness

SOR – 1)

Pivoting in Gaussian elimination – 1),2)

Time marching for PDE's – 1),2),3)

Multigrid – 1),2),3)

MODELS OF COMPUTATION

Predict performance of

algorithms on variety of architectures

different algorithms on same architecture

Permits study & optimization of parameters

Hardware	*Software*
# processors	problem size
memory size	process spawning
communication	process switching
arithmetic	synchronization

Using iterative method for PDE demonstrated weakness of Intel IPSC

Packet size too large

Communication too slow

MAPPING PROBLEM

Map subdivisions of problem onto processors to minimize computation time

OR

Find subdivision that permits optimal mapping for given architecture

Research focused on using graph representations of algorithms & architectures

Efficient algorithms developed to find mappings for specific graphs – e.g. trees onto hypercubes

Technique developed for subdividing domain into subdomains of equal work for adaptive mesh problems

Demonstrated value of hexagonal subdivisions of domain

DYNAMIC LOAD BALANCING

All processors should have equal work

OR

Inefficient use of processors
Longer execution times

Solution: Rebalance

Problem: Not easy

Must account for cost of

Detection
Computation of new distribution
Estimate of performance gain
Redistribution

Work focused on developing and evaluating models of

Computation that can be analysed

COMMUNICATION & SYNCHRONIZATION

Goal: Eliminate *Consequence:* Algorithms fail

Reduction of frequency & amount

Model: SOR for PDE's

Advancing in time independently
Reducing communication cost of global convergence
Reducing frequency of global convergence

Analysis of amount & impact for particular problem mapping on variety of architectures

Model: Adaptive mesh

Measurement of costs for running algorithms

PROGRAMMING ENVIRONMENTS

Force: Set of machine dependent macros that generate Fortran code

Constructs are machine independent
e.g. barrier synchronization

Facilitates programming & portability

Implemented on variety of machines

PISCES: Fortran based virtual machine concept

Provides view of machine appropriate for user's interests

Implemented on VAX, Apollos; to be on Flex/32

BLAZE: New parallel language designed to support scientific programming

Recognition of implicit parallelism

Implemented on VAX, Sun; next: RP3, Flex/32

ACTIVE AREAS OF RESEARCH AT RIACS

(Tying Computer Science to Computing in Science)

ALGORITHMS

- Computational Fluid Dynamics (CFD)
- Computational Chemistry
- Image Contour Extraction

ACCESSIBILITY

- Scientist's Workbench
- Multimedia Mail and Conferencing
- Network Protocols
- Security

LANGUAGES

- Graphical Shell
- Concurrent C

TESTBEDS

- Telescience Testbed
- Center for Advanced Architectures (CAA)

NEW ARCHITECTURES

- Sparse Distributed Memory
- Gas Lattice Automaton

PROJECT: SPARSE DISTRIBUTED MEMORY

CHARACTERISTICS:

- Highly-parallel pattern computer
- Based on Kanerva's mathematical theory of human memory

PARTICIPANTS:

- Pentti Kanerva (RIACS), Project Manager
- Harrison Leong (RIACS)
- David Nagel (Ames)
- Michael Flynn (Stanford)

STATUS:

- Completed simulators
- Building digital prototype

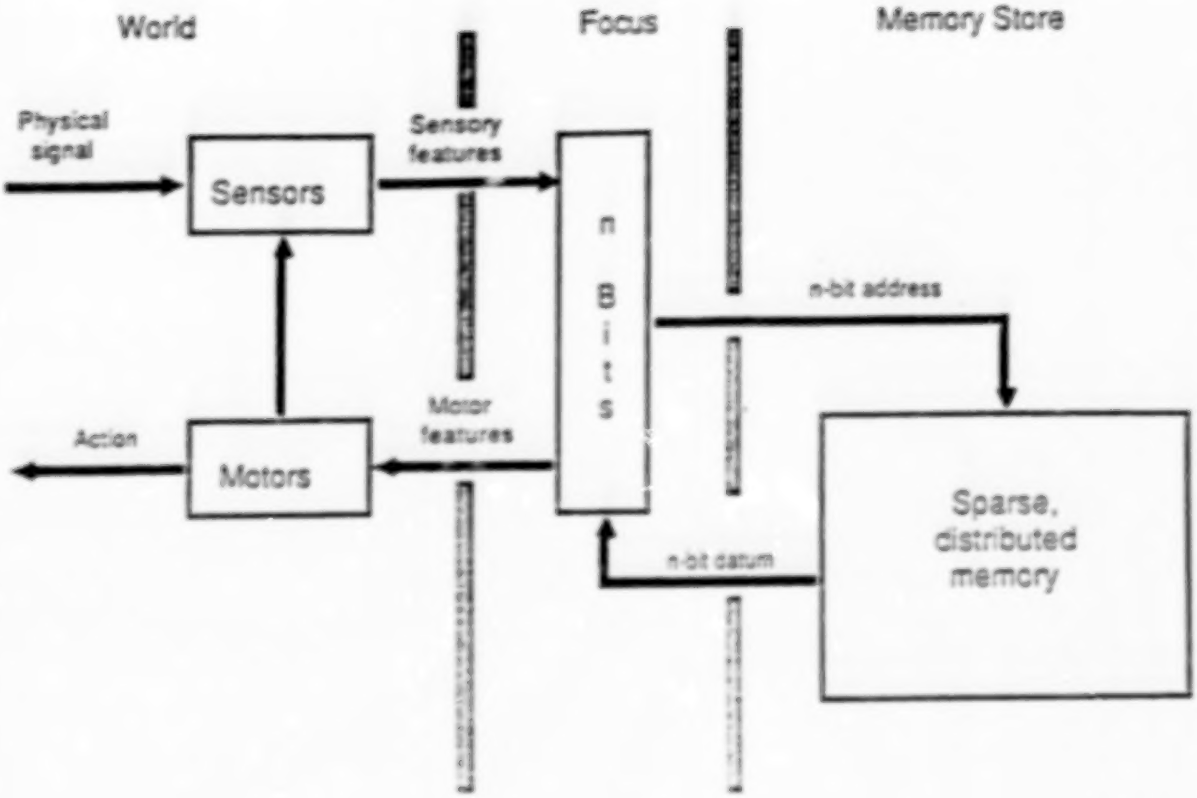
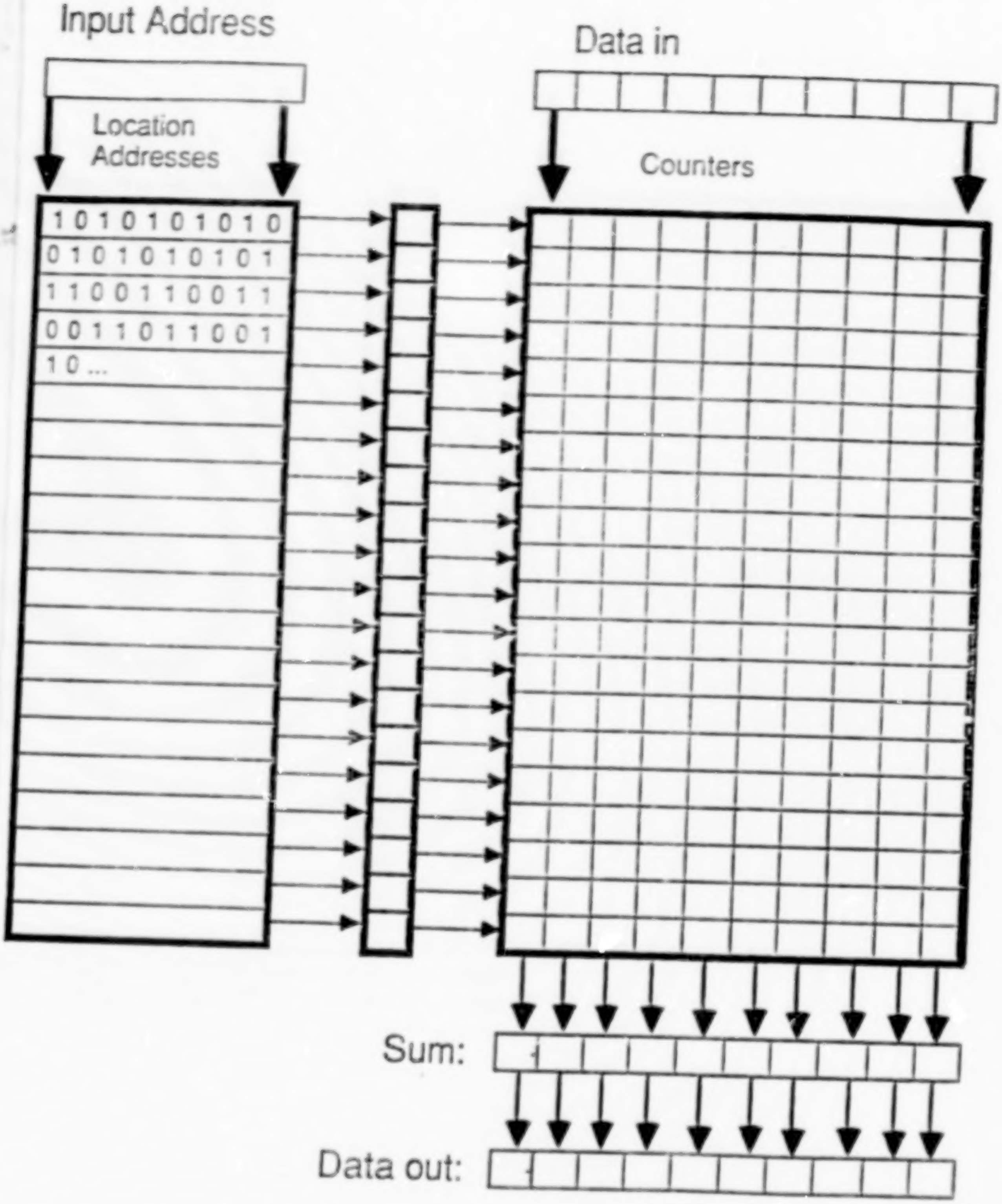

World
Focus
Memory Store
Physical signal
Sensors
Sensory features
n Bits
n-bit address
Action
Motors
Motor features
n-bit datum
Sparse, distributed memory

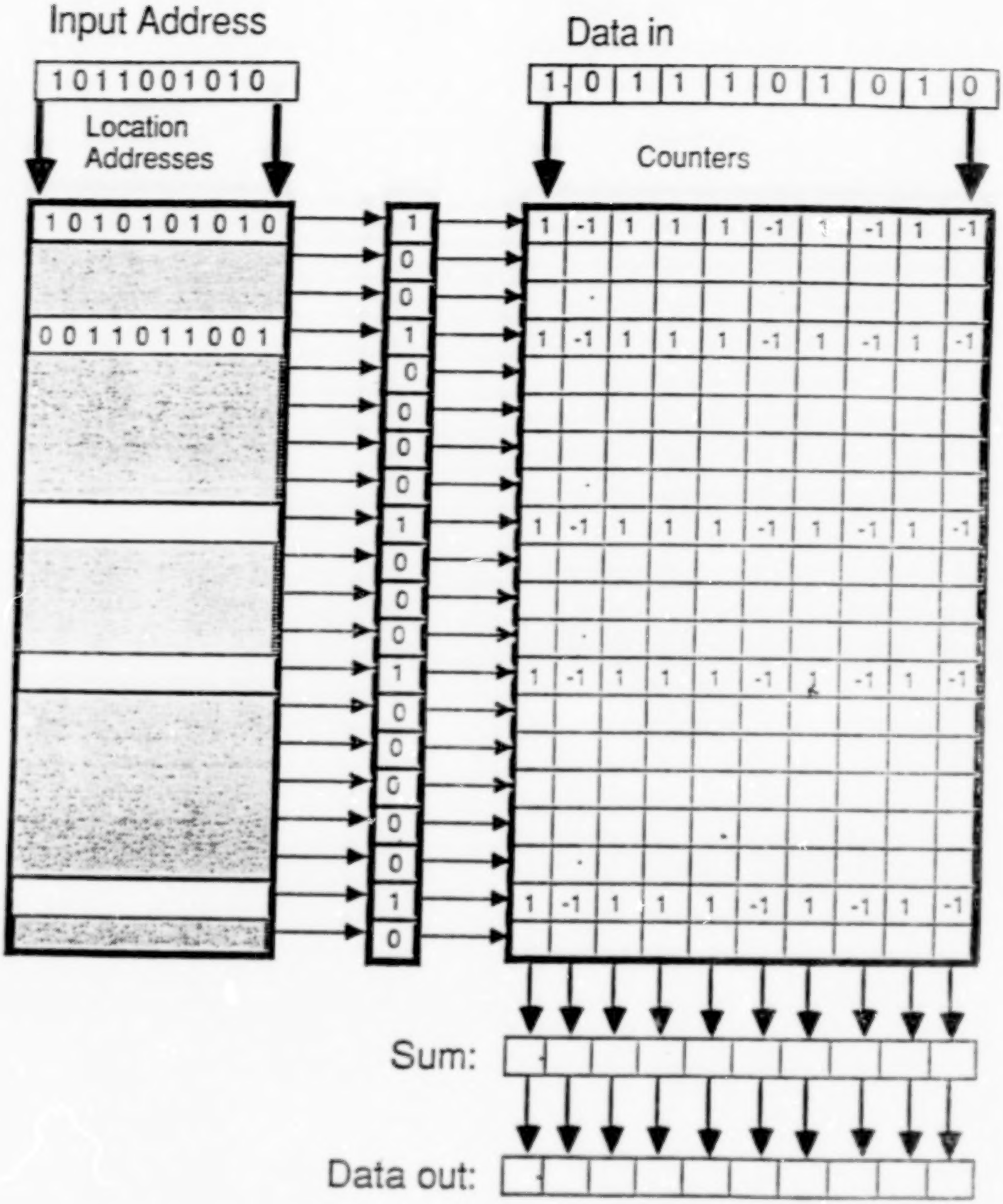
Figure 2. The coupling of the memory to the world. The memory stores a record of the system's past in a form presented to the focus by the system's sensors. The record can include features that control the system's motors. Selective action can be learned by explicit training and by trial and error.

```
Another initial condition? (1-yes)

random (0) or perturbed initial condition(1)?

Which memory state to perturb around?

With what noise level? (0.0 to 1.0)
2
The initial condition is:
   XX  XXXX          XX
X        XXXX        XX
 XXXX    XX      XXXXXXXXXXXX
 XXXXXX  XXXXXXXXXXXXXX  XXXX
 XXXX    XXXXXXXXXXXX      XX
 XXXX    XXXX  XXXXXX      XXXX
XXX    XXXXXX    XXXXXXXXXXXX
     XXXX  XX    XXXXXX  XX
XXXXX  XXXXXXXX  XXXX  XX
X  XX  XXXX    XXXXXXXX    XX
 XXXX    XX         XX     XX
 XXXXXXXXXX  XX  XXXXXX
 XXXX            XXXX  XX  XX
 XX      XX      XXXX
    XXXXXXXXX  XXXXXXXX    XX
     XX
(__tial condition)
type 1 ‹or› to continue

selected locations-11
       XXXX

 XXXX    XXXX    XXXXXXXXXX
 XXXX    XXXXXX  XXXXXXXXXXXX
 XXXX    XXXXXX  XXXX      XX
 XXXX    XXXX  XXXXXX    XXXX
 XXXX    XXXX    XXXXXXXXXXXX
 XXXXXXXXXXXX    XXXXXXXXXX
 XXXXXXXXXXXXXX  XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX      XX
 XXXX    XXXXXX  XXXX
 XXXX    XXXX    XXXX
 XX      XXXX    XXXX
   XX  XXXXXX  XXXXXXXX
```

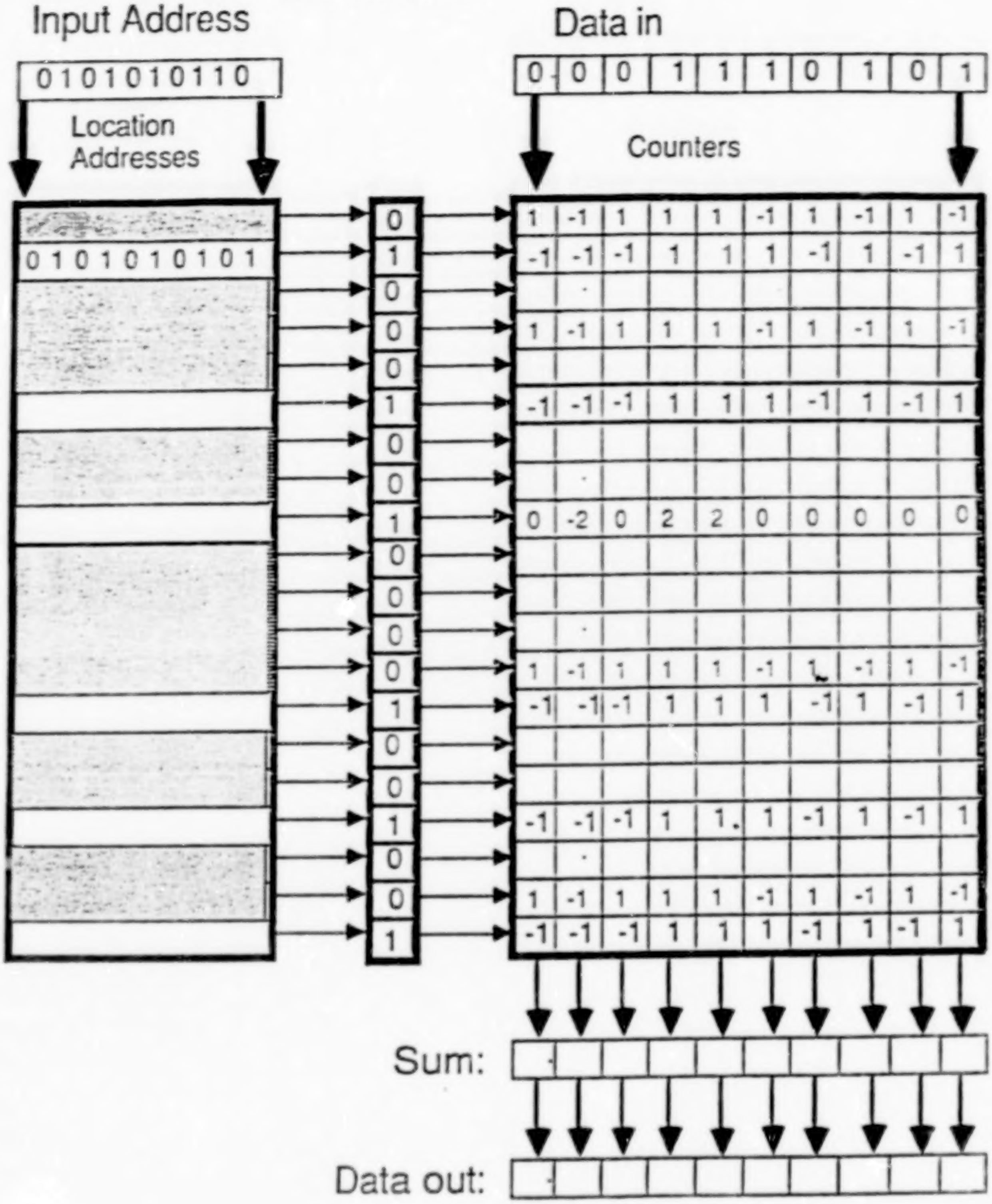
```
 Another iteration with this initial condition (1)?
1
 selected locations-13

 XXXX    XXXX    XXXXXXXXXX
 XXXX    XXXX    XXXXXXXXXXXX
 XXXX    XXXX    XXXX    XXXX
 XXXX    XXXX    XXXX    XXXX
 XXXX    XXXX    XXXXXXXXXXXX
 XXXXXXXXXXXX    XXXXXXXXXX
 XXXXXXXXXXXX    XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX  XXXXXXXX


 Another iteration with this initial condition (1)?
1
 selected locations-14

 XXXX    XXXX    XXXXXXXXXX
 XXXX    XXXX    XXXXXXXXXXXX
 XXXX    XXXX    XXXX    XXXX
 XXXX    XXXX    XXXX    XXXX
 XXXX    XXXX    XXXXXXXXXXXX
 XXXXXXXXXXXX    XXXXXXXXXX
 XXXXXXXXXXXX    XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX    XXXX
 XXXX    XXXX  XXXXXXXX
```

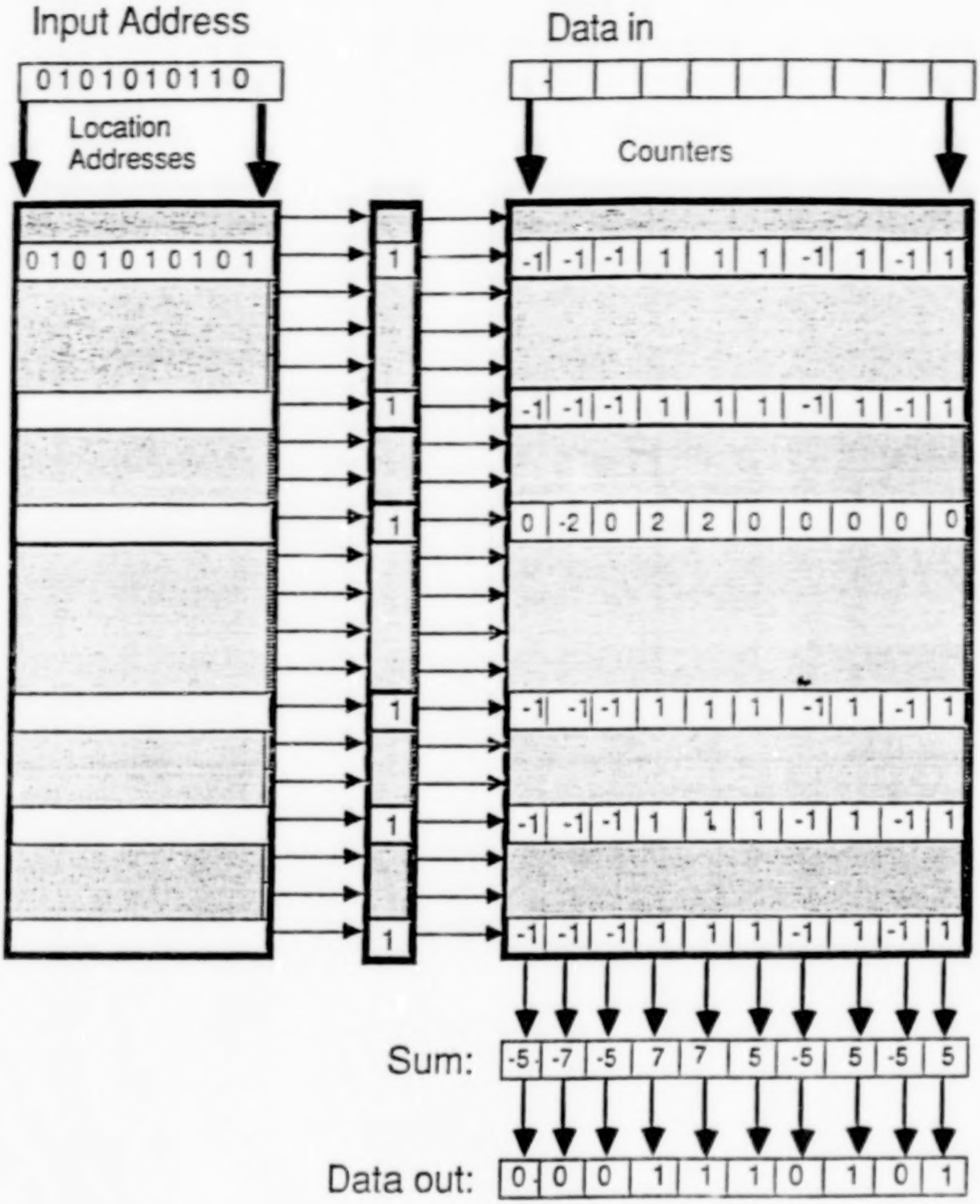
ORIGINAL PAGE IS
OF POOR QUALITY


Input Address
Data in
Location Addresses
Counters
1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 1
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 1 1 0 0 1
1 0 ...
Sum:
Data out:

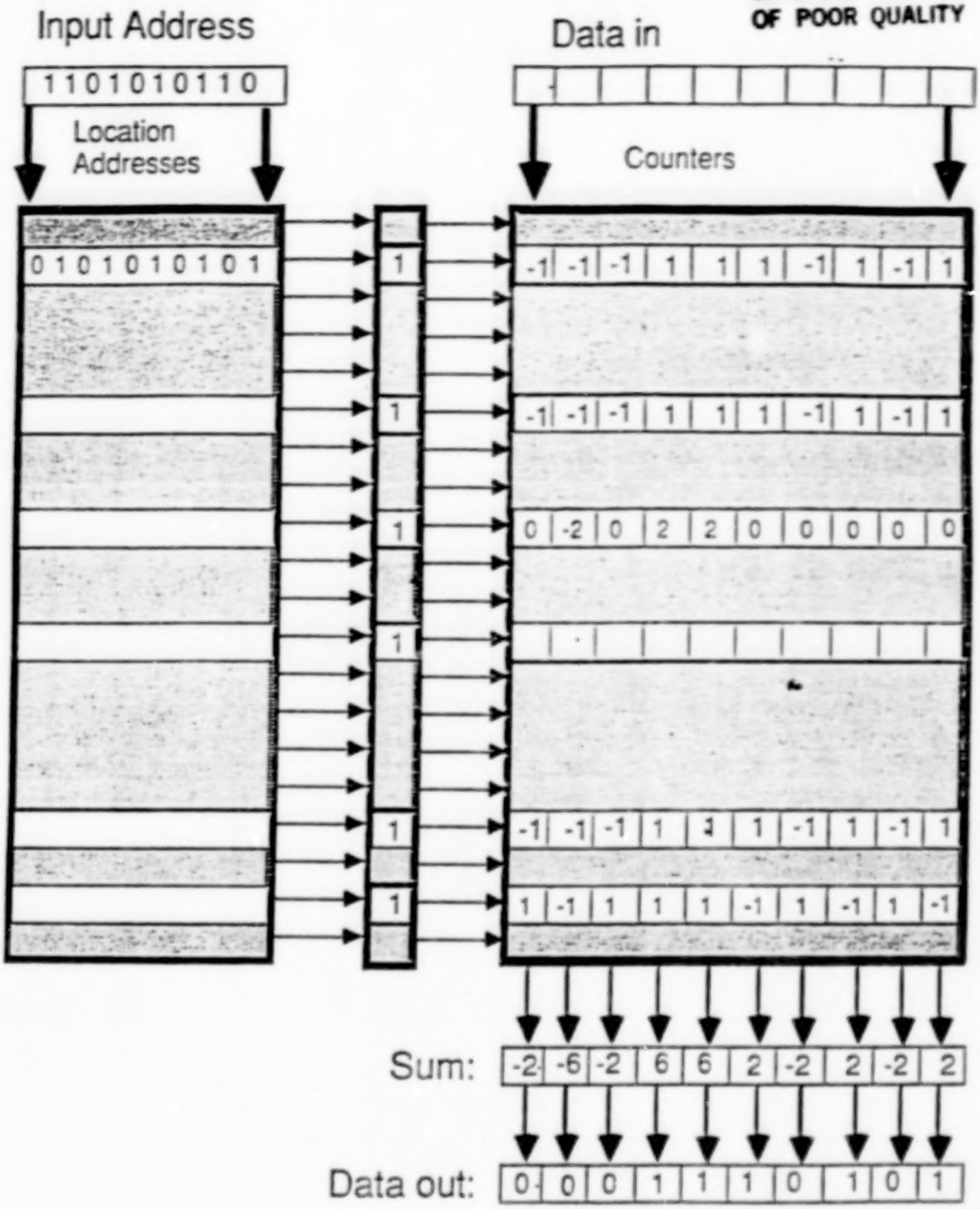

Input Address
1011001010
Location Addresses
Data in
1 0 1 1 1 0 1 0 1 0
Counters
1010101010
0011011001
1 -1 1 1 1 -1 1 -1 1 -1
Sum:
Data out:

ORIGINAL PAGE IS
OF POOR QUALITY


Input Address
0101010110
Location Addresses
0101010101
Data in
0 0 0 1 1 1 0 1 0 1
Counters
0 1 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1
1 -1 1 1 1 -1 1 -1 1 -1
-1 -1 -1 1 1 1 -1 1 -1 1
1 -1 1 1 1 -1 1 -1 1 -1
-1 -1 -1 1 1 1 -1 1 -1 1
0 -2 0 2 2 0 0 0 0 0
1 -1 1 1 1 -1 1 -1 1 -1
-1 -1 -1 1 1 1 -1 1 -1 1
-1 -1 -1 1 1 1 -1 1 -1 1
1 -1 1 1 1 -1 1 -1 1 -1
-1 -1 -1 1 1 1 -1 1 -1 1
Sum:
Data out:


Input Address
0101010110
Location Addresses
Data in
Counters
0101010101
1
-1 -1 -1 1 1 1 -1 1 -1 1
1
-1 -1 -1 1 1 1 -1 1 -1 1
1
0 -2 0 2 2 0 0 0 0 0
1
-1 -1 -1 1 1 1 -1 1 -1 1
1
-1 -1 -1 1 1 1 -1 1 -1 1
1
-1 -1 -1 1 1 1 -1 1 -1 1
Sum: -5 -7 -5 7 7 5 -5 5 -5 5
Data out: 0 0 0 1 1 1 0 1 0 1

ORIGINAL PAGE IS
OF POOR QUALITY


Input Address
1101010110
Location Addresses
Data in
Counters
0101010101
1
-1 -1 -1 1 1 1 -1 1 -1 1
1
-1 -1 -1 1 1 1 -1 1 -1 1
1
0 -2 0 2 2 0 0 0 0 0
1
1
-1 -1 -1 1 1 1 -1 1 -1 1
1
1 -1 1 1 1 -1 1 -1 1 -1
Sum:
-2 -6 -2 6 6 2 -2 2 -2 2
Data out:
0 0 0 1 1 1 0 1 0 1

PROJECT: ADVANCED ALGORITHMS AND ARCHITECTURES

OBJECTIVE:

- Explore and develop new algorithms and architectures for solving large-scale NASA scientific problems

Principals at RIACS:

George Adams, RIACS
John Bruno, UCSB
Tony Chan, UCLA
Merrell Patrick, Duke
Michael Raugh, RIACS (Project Manager)

NASA Collaborators:

David Bailey, NAS
Sanford Davis, Fluid Mechanics
Harvard Lomax, CFD
Harry Partridge, Chemistry
Ken Stevens, Comp. Research

MULTIGRID ALGORITHMS AND HYPERCUBE ARCHITECTURES

(An Advanced Algorithms and Architectures Project)

PARALLEL IMPLEMENTATIONS OF MULTIGRID

- Multigrid optimal for many partial differential equations
- Hypercube architectures near–optimal for multigrid
- Initial studies show promise for CFD

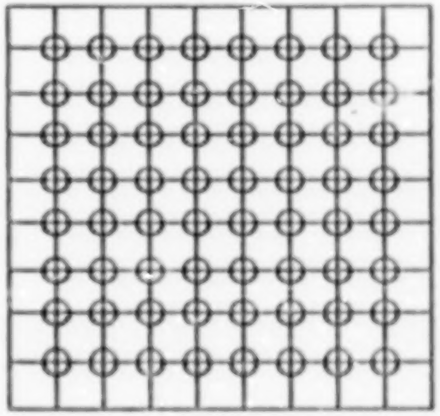
FUTURE DIRECTIONS

- Develop parametric models
- Establish programming principles
- Guide design of hypercube supercomputer for CFD

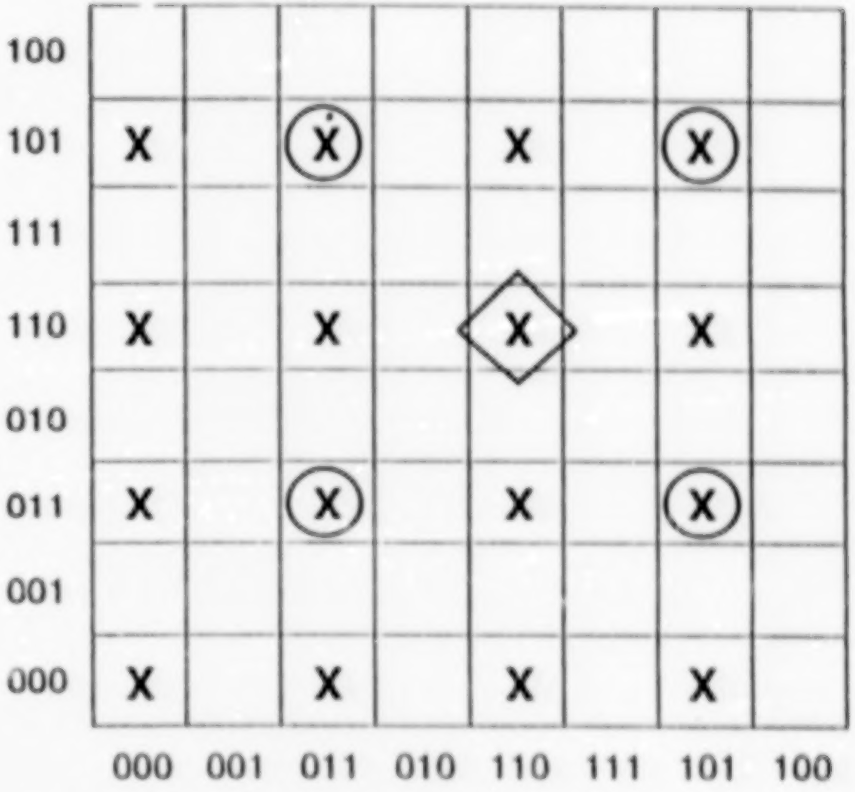
MAPPING OF GRID POINTS TO PROCESSORS USING BINARY REFLECTED GRAY CODE

Model Problem:
Driven Cavity

(Steady State Stream Vorticity Formulation of Navier-Stokes Equations)



Grid Points for 2-D Problem


100
101
111
110
010
011
001
000
X
000
001
011
010
110
111
101
100

Schematic Representation of Hypercube Processors

FFT AND NN–MESH ON HYPERCUBES

(An Advanced Algorithms and Architectures Project)

HYPERCUBE IMPLEMENTATIONS:

- Natural map optimal for FFT but poor for NN–mesh (i–th point mapped to i–th processor)
- Gray code map optimal for NN–mesh (adjacent points mapped to adjacent processors)
- Binary Reflected Gray Code map optimal for NN–mesh and good for FFT

CHAN'S COMMUNICATION MODEL

- Predicts best map for given application
- Example: Use BRGC for NN–mesh applications; shuffle to and from natural map only if more than two FFTs will be performed
- Note: NN–mesh mapping with BRGC also good for Multigrid, Cyclic Reduction, Sorting

IMAGE CONTOUR EXTRACTION

Find boundaries between light and dark regions of unevenly lit image.

Break image into regions: a) compute thresholds for each region,
b) reconcile boundaries.

Code runs on Sequent Balance 21000 and on Intel hypercube.

How to partition into regions to balance load in Phase (a)?

How to reconcile boundaries to balance communication in Phase (b)?

What performance monitoring tools aid the programming of these phases?

How fast can image contours be extracted by this method?

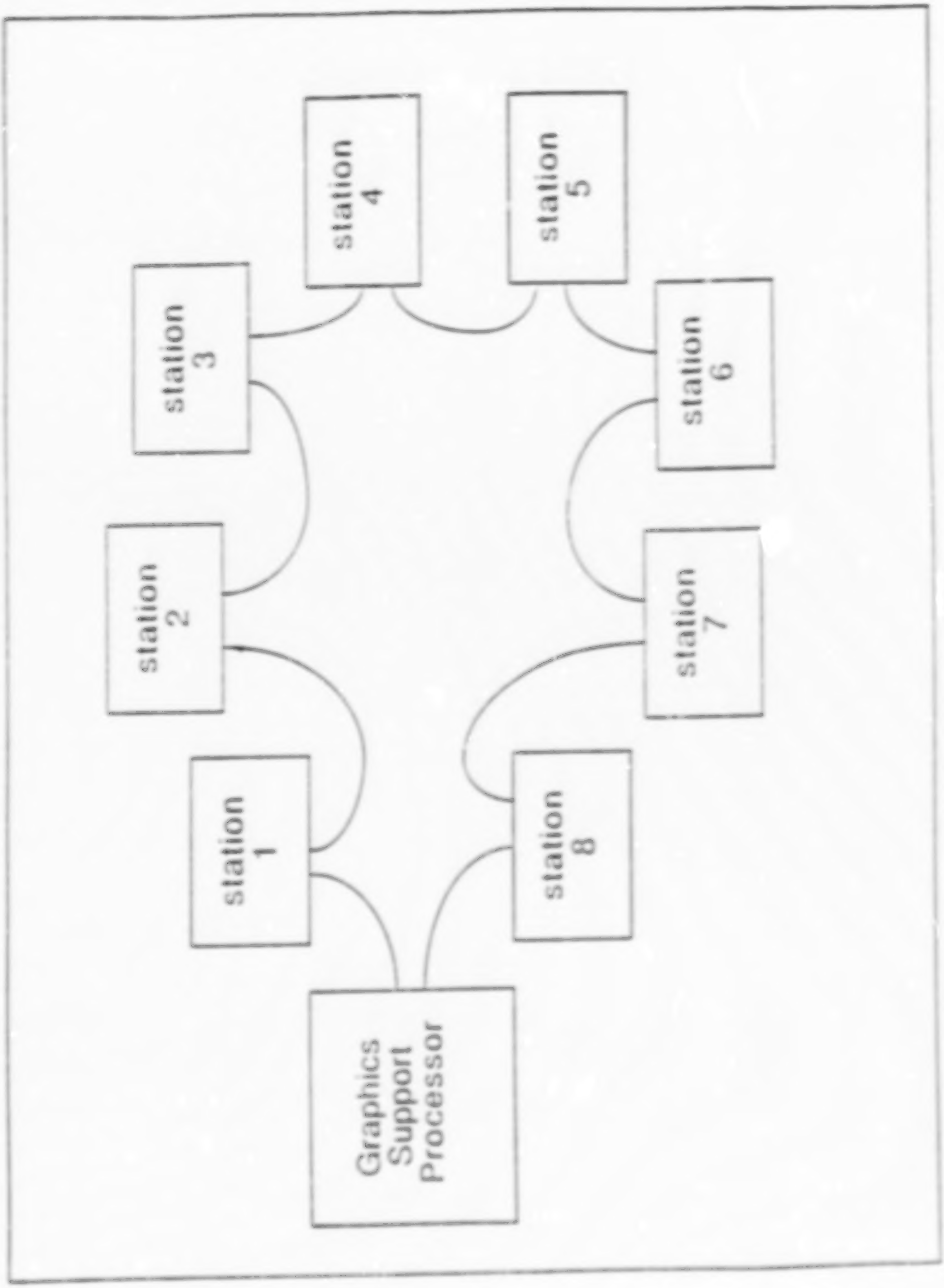
NETWORK PROTOCOLS

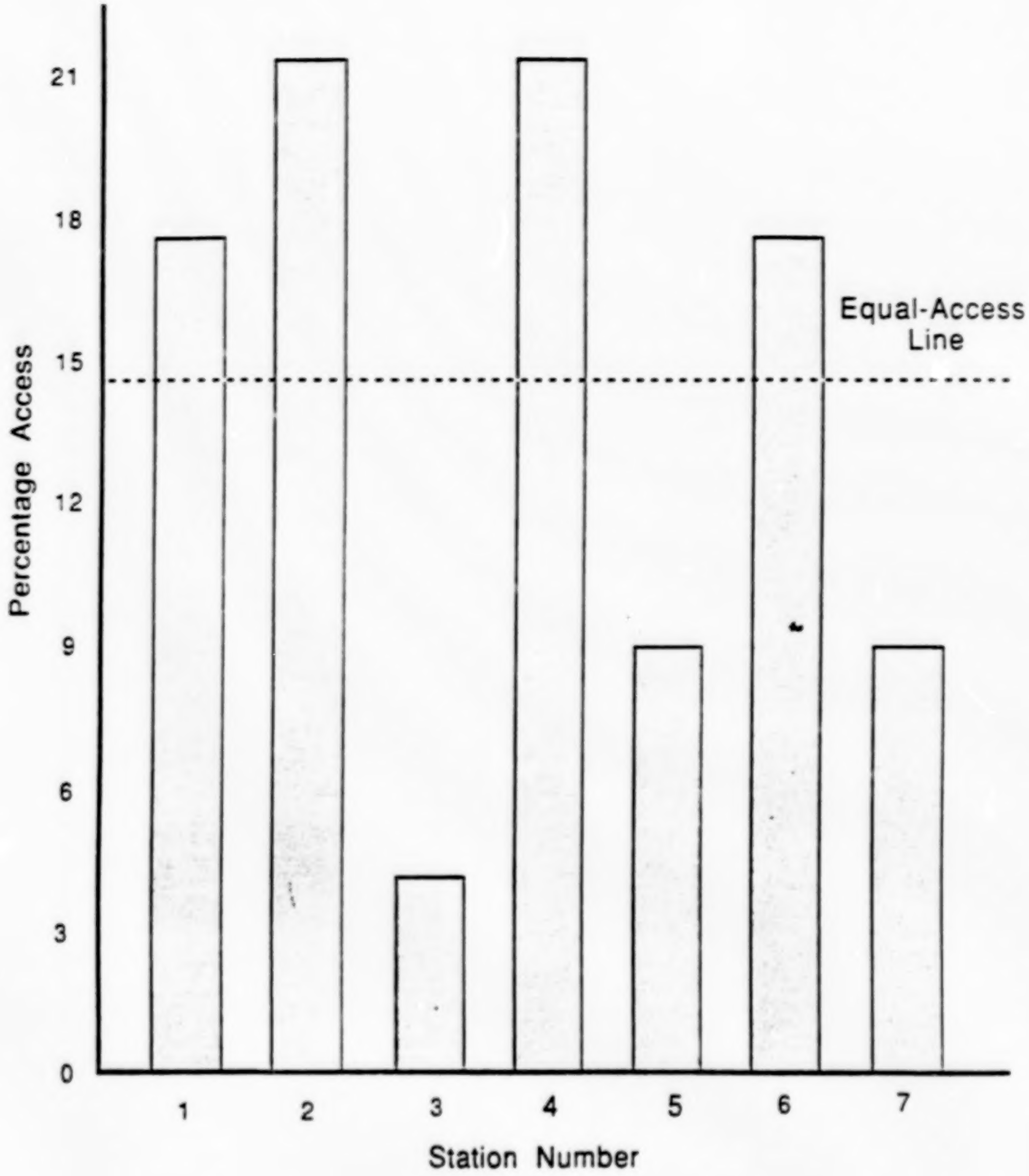
- Fiber Distributed Data Interface (FDDI) is an ANSI draft proposed standard for a 100 Megabit per second fiber-optic token ring
- FDDI is a candidate media access protocol for the Space Station
- Complexity of FDDI timed token protocol necessitates analytic justification of performance claims presented in standards documents
- Several claims have been analytically established as a result of this project
- Also have shown that, under some circumstances, the FDDI claim of equal access to all stations on the ring is not valid

The next two figures pertain to the FDDI protocol study.

The first figure shows 8 workstations connected on a ring to a graphics processor. The FDDI protocol could be used if the ring is fiber optic.

The second figure shows the results of a simulation study of FDDI. Under certain (reproducible) load conditions, where message length is long enough to delay the token at some stations, the stations do not obtain equal access to the ring. FDDI parameters must be set within allowable ranges to guarantee the equal-access property.


Graphics Support Processor
station 1
station 2
station 3
station 4
station 5
station 6
station 7
station 8



Fairness of Channel Access for FDDI High Bandwidth Token Ring

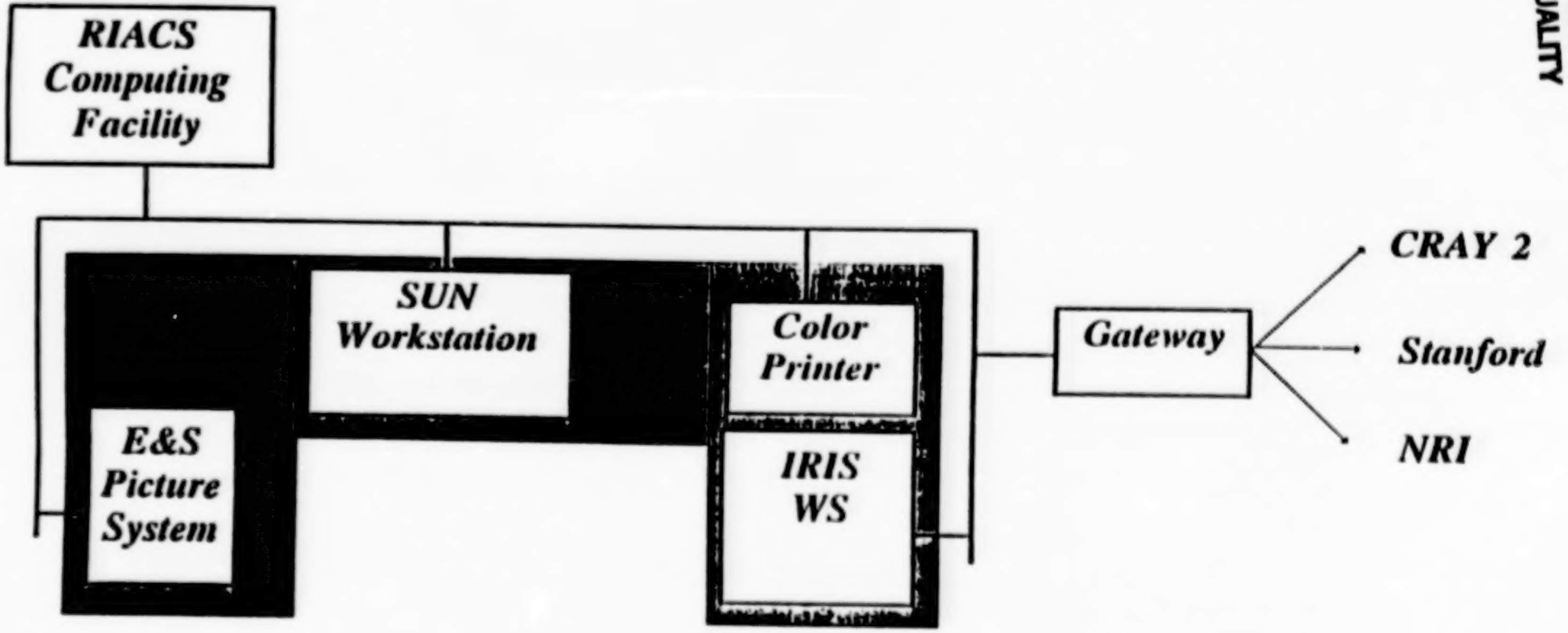
C-3

SCIENTIST'S WORKBENCH

- Support scientific investigation from problem formulation through computation solutions to publication
- Initial step towards *Scientist's Aide*
- In collaboration with working scientists:
    - Develop usage scenarios
    - Define functions needed (e.g. graphics, software development environment, remote control of supercomputations, symbolic math)
    - Integrate hardware and software
    - Use in advanced scientific research
- Iterative process

INITIAL SCIENTIST'S WORKBENCH

ORIGINAL PAGE IS
OF POOR QUALITY


RIACS Computing Facility
SUN Workstation
E&S Picture System
Color Printer
IRIS WS
Gateway
CRAY 2
Stanford
NRI

MULTIMEDIA CONFERENCING AND COLLABORATIVE RESEARCH

- Networking and Workstations Provide Tools to Support Collaborative Research
- Computer Supported Real Time Multimedia Conferencing Available in Prototype
- Questions as to Required Features and Best Ways to Utilize
    - Use Scientist's Workbench Environment to Evaluate
    - Collaborative with Stanford and SRI

INTERAGENCY RESEARCH INTERNET

- Interconnection of Agency Funded Networks to Allow Cost-Effective Sharing of Network and Computing Resources, and Facilitate Collaborative Research
- RIACS Assisting FCCSET Network Working Group to Plan and Implement the Interconnection

TELESCIENCE TESTBED

- Need for Rapid Prototyping Capability to Evaluate Advanced Information Technologies and Their Role in Space Station and the Space Station Information System
- RIACS Heading Working Group to Plan for Such an Environment
- Proposal being Drafted to Code E
  - Ten Universities Involved to Form Distributed Testbed
  - RIACS to Organize and Manage

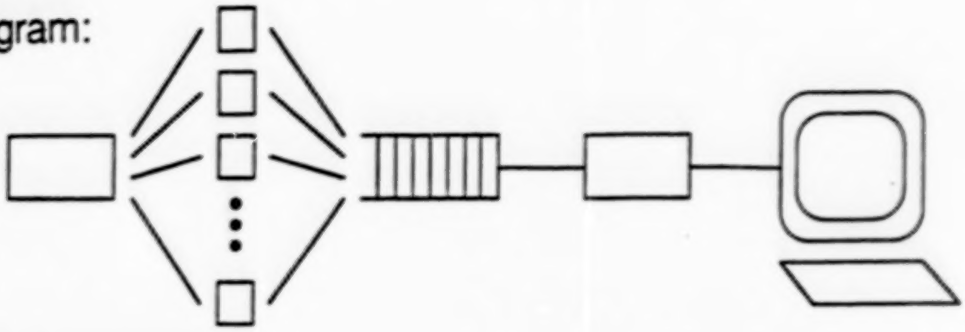
This slide depicts a style and language for writing parallel programs. The top third is a process-level block diagram of a typical concurrent program. This shows an array of computation processes (vertical stack of boxes) receiving pieces of work from a ticket server and feeding results to a queue manager for subsequent display.

The middle portion shows some of the program to implement the block diagram. The language is Concurrent C, developed at AT&T Bell Labs and implemented in a translator at RIACS.

The third portion shows how this concurrent program "plugs into" and interactive graphics application running on a color workstation. The inputs to the program are provided from a user-interface "control panel" and the output drives a graphics window.

Parallel Programming Languages

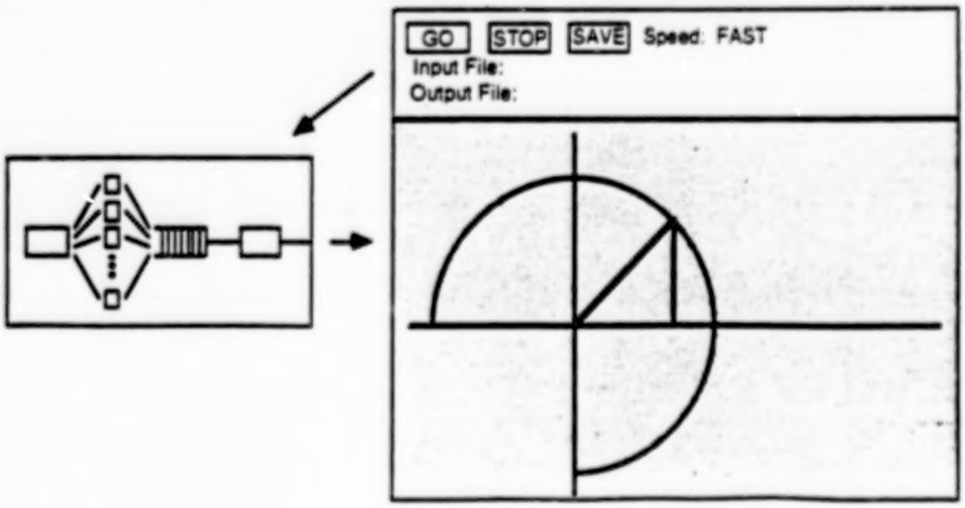
Block Diagram:



Concurrent C program:

```
ticket = create ticket(N);
for ( i=0 ; i<N i ;++ )
        compute[i] = create compute(i,ticket,display);
...

process body compute ( i, tick, disp)
{
       while ( tick.next < N ) {
            ...
             disp.put();
       }
}
```

Interactive Graphics-oriented Application:


GO
STOP
SAVE
Speed: FAST
Input File:
Output File:

A PROPOSAL TO ESTABLISH

A
NASA CENTER OF EXCELLENCE
IN
SPACE DATA INFORMATION SCIENCES
(CESDIS)

NOVEMBER 17, 1986

NASA/GSFC

OVERVIEW

- OSSA DATA PROBLEMS
- THE USRA/U. MD. PROPOSAL, MANAGEMENT, BUDGET, ISSUES
- COMMENTS AND RECOMMENDATIONS

FUTURE INSTRUMENTS AND THEIR DATA RATES (MB/s)*

SAR - 300 MB/s

HIRIS - 160 MB/s

TM/MLA - 85 MB/s

TIMS - 30 MB/s

SIRTF - 10 MB/s

MODIS - 10 MB/s

HRSO - 5 MB/s

FUSE - 2 MB/s

SP.TELE.- .2 MB/s (800 GB/YR) OR (400 O.D. PLATTERS)

TOTAL 603 MB/s OR 50.000 TAPES/DAY OR
20-YEAR NSSDC HOLDINGS

* 1 MB/s = 75 TAPES/DAY AT 6250 BPI OR 27.375 TAPES/YEAR

OBSERVING SYSTEMS OVERWHELMING INFORMATION PROCESSING SYSTEMS

DATA VOLUME:	SENSOR DATA RATES GROWING $\sim N^4$ (10^{13}-10^{15} B/YR); HIGH SPEED STORAGE/ACCESS TECHNOLOGIES GROWING (10^{9}-10^{12} B/YR); TYPICAL SCIENCE SCENARIO NEEDS TO HANDLE 1000 TAPES/INSTRUMENT
INFORMATION SYSTEMS:	DISTRIBUTED MASSIVE ARCHIVES, HETEROGENEOUS ENVIRONMENTS, MULTI-SPECTRAL/SPATIAL DATA STREAMS, COORDINATED OBSERVATIONAL CAMPAIGNS
COMPUTATIONAL ANALYSIS:	SATELLITE MEASUREMENTS REQUIRE COMPLEX INTERPLAY BETWEEN DATA AND THEORETICAL MODELS, MULTI-DISCIPLINARY SYSTEMS, INTERACTIVE IMAGE AND GRAPHICS SYSTEMS, MELDING VERY LARGE INFORMATION BASES

----- ----- ----- ----- ----- ----- ----- ----- -----

NASA ABILITY TO DEAL WITH THE DATA SITUATION RAISING NUMEROUS CONCERNS IN SCIENCE COMMUNITY

WHAT IS NASA DOING ABOUT DATA PROBLEMS?

- o COLLECTING LIMITED DATA SAMPLES FROM HIGH DATA RATE INSTRUMENTS (DUTY CYCLES)
- o STORING AND PERMANENTLY ARCHIVING ALL AVAILABLE DATA UNTIL TECHNOLOGY CATCHES UP
- o DEVELOPING MULTIPLE PILOT DISCIPLINE DATA SYSTEMS TO ADDRESS CURRENT DATA PROBLEMS
- o CONTRACTING OUT FOR DATA SYSTEMS BASED ON EXISTING TECHNOLOGIES TO MEET SPECIFIC FLIGHT PROGRAMS

DIFFICULTIES IN DEALING WITH SPACE SCIENCE DATA PROBLEMS STEM FROM:

o SEVERE LACK OF SKILLED, IN-HOUSE EXPERTISE

  - NASA'S SCIENTIFIC EXPERTISE DOES NOT ADEQUATELY ENCOMPASS HIGHLY SPECIALIZED DATA SYSTEMS AND TECHNOLOGIES

  - FLOOD OF COMPETITIVE, PROFESSIONALLY EXCITING POSITIONS (NSF, DOD AND INDUSTRY/STATE CONSORTIUMS) SERIOUSLY IMPACTING ABILITY TO RECRUIT OR HOLD FIRST-RATE TALENT

o UNIVERSITY COMPUTATIONAL SCIENCES NOT ADDRESSING FUTURE OSSA DATA PROBLEMS

  - NASA'S SPACE DATA PROBLEMS UNKNOWN IN MOST COMPUTER SCIENCE DEPARTMENTS

  - LITTLE COMMERCIAL PAY-OFF TO INDUSTRY FOR NASA-UNIQUE PROBLEMS

o EXTREMELY COMPLEX PROBLEM CUTTING ACROSS SCIENTIFIC BOUNDARIES WITH IMMATURE EMERGING TECHNOLOGIES

HOW DO WE SOLVE THE DATA PROBLEMS?

WE NEED TO:

- o NEED TO BRING TO BEAR ADVANCED TECHNOLOGIES AND STATE-OF-THE-ART SYSTEMS EQUAL TO DEALING WITH THE CHALLENGE

  (MEANS ACQUIRING STATE-OF-THE-ART MASS STORAGE PROCESSING SYSTEMS, DEVELOPING AUTOMATED INTELLIGENT ANALYSIS SYSTEMS, ETC.)

- o FOCUS UNIVERSITY AND INDUSTRY INFORMATION SCIENCE RESEARCH ONTO NASA'S RELATED DATA PROBLEMS

- o SELECTIVELY BROADEN OUR LIMITED POOL OF TALENT IN COMPUTATIONAL AND INFORMATION SCIENCES

- o CONTINUE SUPPORTING THE PILOT DATA SYSTEMS PROGRAMS AND OTHER DATA TECHNOLOGY TEST BEDDING STUDIES

- o BEGIN BY MAKING A LONG-TERM RESEARCH INVESTMENT IN THE INFORMATION SCIENCES TAILORED TO OSSA'S DATA PROBLEMS

NASA/GSFC HAS RECEIVED AN

INNOVATIVE PROPOSAL FROM USRA AND THE U. OF MARYLAND

TO ESTABLISH A

CENTER OF EXCELLENCE IN

SPACE DATA INFORMATION SCIENCES

(CESDIS)

TO DEAL WITH NASA'S LONG-TERM DATA PROBLEMS

CESDIS PROPOSES TO:

- o ESTABLISH A "CONSORTIUM" OF ACADEMIA AND INDUSTRY-
  - COMMITTED TO SPECIALIZED INFORMATION SCIENCE RESEARCH TAILORED TO NASA'S LONG-TERM SPACE DATA PROBLEMS

- o CREATE "CRITICAL MASS EFFECT" BY-
  - UNITING A LARGE NUMBER OF UNIVERSITY AND INDUSTRY EXPERTS IN COMPUTER SCIENCE, DATA SYSTEMS IMAGE AND SIGNAL PROCESSING TO WORK WITH NASA SCIENTISTS ON SPACE DATA PROBLEMS

- o UTILIZE "HIGH TECH" RESOURCES-
  - MAKE AVAILABLE TO CESDIS INVESTIGATORS ACCESS TO NASA SUPERCOMPUTERS, NETWORKS, PARALLEL PROCESSING SYSTEMS, MEGA DATA SYSTEMS, IMAGE ANALYSIS SYSTEMS, ROBOTS, ETC.

THE CENTER OF EXCELLENCE WILL:

- HOUSE AT GSFC A "TELE-INSTITUTE" WITH MEMBERS NETWORKED FROM THEIR HOME INSTITUTIONS TO EACH OTHER AND APPROPRIATE NASA CENTERS
- ENCOURAGE SUBSTANTIAL ON-SITE INTERACTIONS WITH THE MOST QUALIFIED ACADEMIC AND INDUSTRIAL EXPERTS IN THE COUNTRY
- PROJECT NASA COMPUTATIONAL RESEARCH INTERESTS AND NEEDS INTO UNIVERSITY DEPARTMENTS AND INDUSTRY IR&D
- ESTABLISH WORKING-LEVEL TIES WITH UNIVERSITY GRADUATE STUDENTS AND FACULTY TO ENSURE A POOL OF TALENT FOR NASA'S FUTURE
- CREATE A FOCUS OF RESEARCH EXCELLENCE IN INFORMATION SCIENCE AT GSFC

USRA/UNIVERSITY OF MARYLAND PROPOSAL HIGHLIGHTS:

- ADMINISTRATION BY USRA (FUNDING AND SUBCONTRACTS), MANAGED BY DIRECTOR, U/MD
- DEPARTMENT OF COMPUTER SCIENCES UNIVERSITY OF MARYLAND WILL ESTABLISH FULL TENURED PROFESSORSHIP FOR DIRECTOR; APPOINTMENT SUBJECT TO CONCURRENCE OF NASA; UNIVERSITY OF MARYLAND PROVIDES 50% OF DIRECTOR'S ACADEMIC YEAR SALARY
- NASA-SUPPORTED FULL FACULTY TEACHING APPOINTMENTS AT U.S. UNIVERSITIES; APPROXIMATELY 40% OF YEAR ON-SITE AT GODDARD; NASA/UNIVERSITY SHARED 5-YEAR ROLLING APPOINTMENTS
- INDUSTRY PARTICIPATION ON EQUAL BASIS WITH UNIVERSITIES WITH NO SALARIES, NO PROPRIETARY RIGHTS TO RESEARCH RESULTS
- UNIVERSITY AND INDUSTRY PARTICIPATION THROUGH PEER-REVIEWED PROPOSAL PROCESS
- CESDIS ADVISORY BOARD CONSISTING OF ALL COOPERATING UNIVERSITIES, INDUSTRIAL MEMBERS (2/3) AND GOVERNMENT (NASA OR OTHERS) (1/3) FOR SCIENCE OVERSIGHT
- ACCESS TO NASA/GSFC COMPUTING RESOURCES

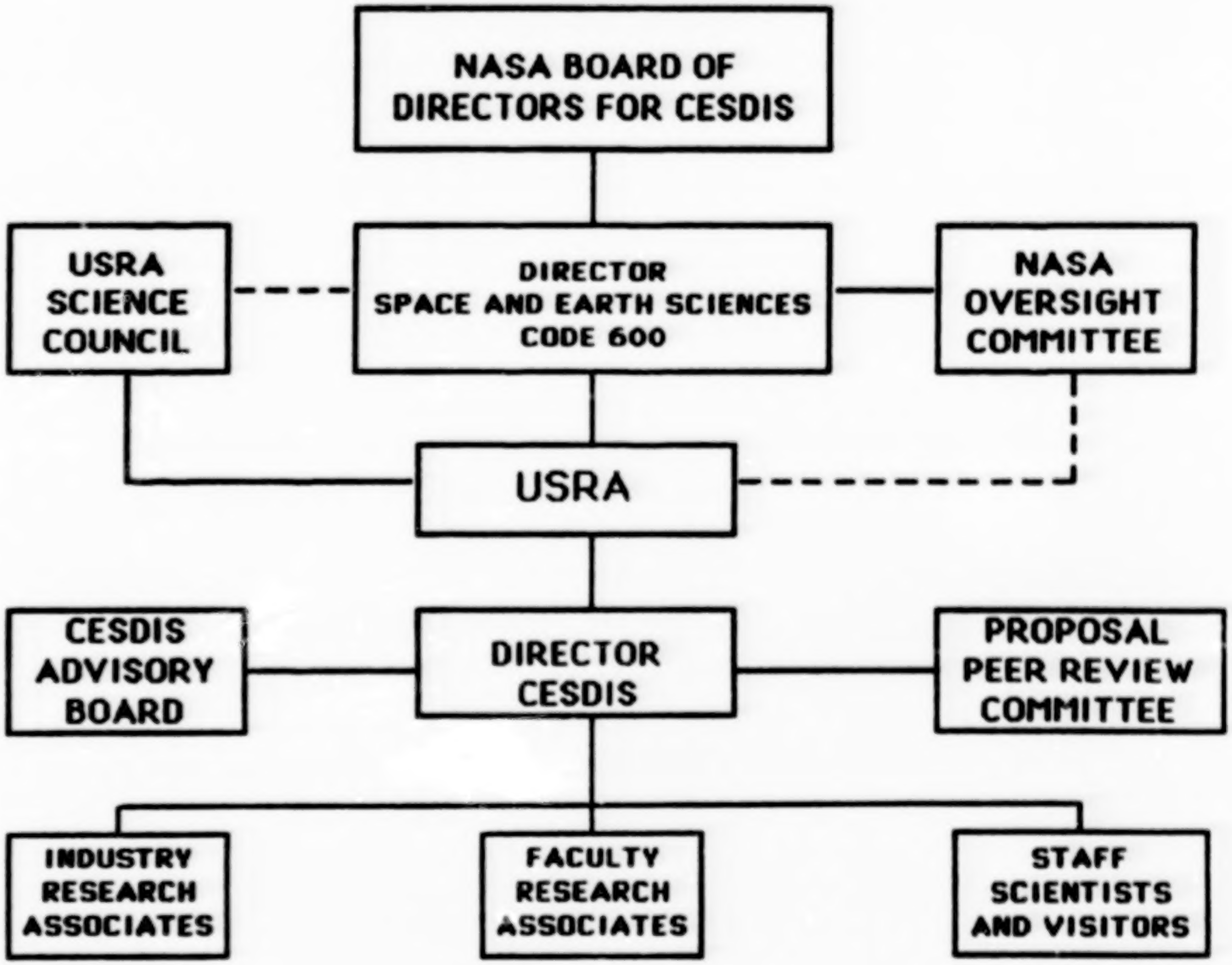
UNIVERSITY MEMBERSHIP SELECTION PROCESS

- DIRECTOR OF CESDIS IDENTIFIES RESEARCH PROGRAM AREAS WITH CONCURRENCE OF NASA BOARD OF DIRECTORS
- USRA ISSUES CALL FOR PROPOSALS
- DIRECTOR OF CESDIS ESTABLISHES PEER REVIEW COMMITTEE--UNIVERSITY/INDUSTRIAL/GOV'T.
- PEER REVIEW COMMITTEE RANKS PROPOSALS -
  - QUALITY
  - INSTITUTIONAL EXPERTISE
  - UNIVERSITY COMMITMENT
- SELECTION BY DIRECTOR, CESDIS, CONCURRENCE OF NASA OVERSIGHT COMMITTEE
- SUBCONTRACT(S) AWARDED BY USRA, GSFC TECHNICAL MONITORS

POTENTIAL RESEARCH AREAS

- MASSIVE DATA BASE SYSTEMS AND ARCHITECTURE STUDIES
- IMAGE DATA COMPRESSION FOR GROUND AND ON-BOARD PROCESSING
- HIGH SPEED (10^2-10^3 Mb/S) ON-LINE DATA ACCESS SYSTEMS
- CONCURRENT PROCESSING APPLICATIONS
- RESEARCH MISSION DATA SYSTEMS SUPPORT: ST, HRSO, MODIS, EOS
- PATTERN RECOGNITION FOR INFORMATION ANALYSIS
- EXPERT SYSTEMS AND NATURAL QUERYING LANGUAGES
- ROBOTICS AND VISION

OPERATING RELATIONSHIPS



NASA BOARD OF DIRECTORS FOR CESDIS

USRA SCIENCE COUNCIL

DIRECTOR SPACE AND EARTH SCIENCES CODE 600

NASA OVERSIGHT COMMITTEE

USRA

CESDIS ADVISORY BOARD

DIRECTOR CESDIS

PROPOSAL PEER REVIEW COMMITTEE

INDUSTRY RESEARCH ASSOCIATES

FACULTY RESEARCH ASSOCIATES

STAFF SCIENTISTS AND VISITORS

OPERATING CONCEPT

UNITS	COMPOSITION	FUNCTIONS
NASA BOARD OF DIRECTORS FOR CESDIS	- HEADQUARTERS AA's (E, R) - GSFC DIRECTORS OF (600, 700, 500) - OTHER NASA CENTERS - NSF - CHAIRED, GSFC DEPUTY DIRECTOR	- APPROVE RESEARCH AREAS, PROGRAM PLANS, POLICIES - GENERAL REVIEW OF ACCOMPLISHMENTS, ISSUES, BUDGETS
NASA OVERSIGHT COMMITTEE	- GSFC LABORATORY AND DIVISION CHIEFS - HEADQUARTERS PROGRAM MANAGERS - OTHER NASA CENTERS, DIVISION CHIEFS - CHAIRED, CHIEF, SPACE DATA & COMPUTING DIVISION, GSFC	- RECOMMENDS RESEARCH AREAS, POLICIES - REVIEWS PROGRAMS, PROGRESS, VISITOR ACTIVITIES, WORKSHOPS, LECTURES
CESDIS ADVISORY BOARD	- PARTICIPATING UNIVERSITY AND INDUSTRY DEPARTMENT HEADS - AD HOC NASA REPRESENTATIVES (NASA AND OTHER FEDERAL AGENCIES - CHAIRED BY ELECTED MEMBER	- GUIDANCE ON MANAGEMENT, OPERATIONS WITH PARTICIPATING MEMBERS
CESDIS PROPOSAL PEER REVIEW COMMITTEE	- UNIVERSITIES, INDUSTRY, GOV'T - SELECTED BY CESDIS DIRECTOR	- REVIEW AND RANK PROPOSALS - RECOMMEND TO CESDIS DIRECTOR
USRA SCIENCE COUNCIL	- STANDING USRA COMMITTEE - AD HOC NASA MEMBERSHIP	- CONDUCTS PERIODIC REVIEW OF PERFORMANCE, RELATIVE CONTRIBUTIONS - REPORTS TO USRA BOARD OF TRUSTEES - PROVIDE NASA SPECIFIC WRITTEN APPRAISALS - CONDUCT TRIENNIAL PEER REVIEW, AS REQUESTED

EXAMPLES OF POLICY/BUDGET ISSUES

- ACCEPTS NO MORE THAN X% NON-NASA FUNDING SUPPORT
- TOTAL SUPPORT NOT MORE THAN Y% OF BASE INSTITUTIONAL SUPPORT
- VISITING MEMBERS NOT MORE THAN TWO YEARS
- FRA's SHALL ACCEPT NO MORE THAN XX% NON-NASA SUPPORT
- WHAT PERCENTAGE OF RESOURCES ALLOCATED TO CESDIS, FRA's, IRA's?
- CAN CESDIS RELEASE ANY SOFTWARE AND HARDWARE DESIGNS TO INDUSTRIAL MEMBERS?
- RELATIVE EFFORTS BETWEEN INFORMATION SCIENCE AND INFORMATION APPLICATIONS

USRA AND GSFC ESTABLISHED A PLANNING COMMITTEE

MEMBERSHIP

VICTOR BASILI	UNIVERSITY OF MARYLAND
SHELTON ALEXANDER	PENNSYLVANIA STATE UNIVERSITY
BRUCE ARDEN	PRINCETON UNIVERSITY
PETER BANKS	STANFORD UNIVERISTY
RALPH BERNSTEIN	IBM
ROBERT SPEARING	GODDARD SPACE FLIGHT CENTER
MOUSTAFA CHAHINE	JET PROPULSION LABORATORY
PETER DENNING	RIACS
BERNARD GALLER (CHAIR)	UNIVERSITY OF MICHIGAN
ANTHONY HEARN	RAND CORPORATION
EUGENE ISAACSON	NEW YORK UNIVERSITY
LARRY JODSAAS	CONTROL DATA CORPORATION
DONALD JOHNSON	UNIVERSITY OF WISCONSIN
ROBERT DORFMAN	UNIVERSITY OF MARYLAND
ROBERT KURUCZ	SMITHSONIAN ASTROPHYSICAL OBSERVATORY
LOUIS LANZEROTTI	AT&T (BELL LABS)
ROBERT McPHERRON	UNIVERSITY OF CALIFORNIA AT LOS ANGELES
JOHN QUANN	GODDARD SPACE FLIGHT CENTER
ETHAN SCHREIER	SPACE TELESCOPE SCIENCE INSTITUTE
GERALD SOFFEN	GODDARD SPACE FLIGHT CENTER
JAMES TRAINOR	GODDARD SPACE FLIGHT CENTER
MIKE McGREEVY	NASA/HQ/OAST
CALDWELL McCOY	NASA/HQ/OSSA

GSFC DISCUSSIONS WITH ACADEMIC AND INDUSTRY LEADERS AND THE CONCLUSIONS OF THE PLANNING COMMITTEE INDICATE:

COMMITTEE:

- "HIGHLY INNOVATIVE AND HOLDS GREAT PROMISE FOR IMPROVING NASA SCIENTIFIC DATA AND INFORMATION SCIENCE PROGRAM"
- "PROPOSAL MAKES SENSE IN SEVERAL WAYS:
  - o ADDRESSES THE PROBLEM OF SCARCITY OF COMPUTER SCIENTISTS
  - o CONSISTENT WITH EMERGING TECHNOLOGICAL CAPABILITIES
  - o TAKES ADVANTAGE OF LOCAL STRENGTH OF UNIV. OF MARYLAND
  - o DEEPENS INTERACTION BETWEEN GSFC AND COMPUTER SCIENCE RESEARCH ACTIVITIES AT OTHER UNIVERSITIES AND INDUSTRY."

ACADEMIC AND INDUSTRY:

- INTEREST EXPRESSED BY COLUMBIA, HARVARD, MIT, NYU, U/VA, RPI, GEORGE MASON, U/MD, JOHNS HOPKINS, CONTROL DATA, AND DIGITAL EQUIPMENT CORPORATION (DEC)

WHAT'S ADVANTAGE TO GSFC?

CODE 600:	DIRECT SUPPORT TO ALL THE LABORATORIES THROUGH ADVANCED ANALYSIS TECHNIQUES AND SYSTEMS, DATA BASE SYSTEMS, DATA MANIPULATION, ADVANCED SOFTWARE SYSTEMS AND MISSION ANALYSIS SUPPORT. INTERDISCIPLINARY SCIENCE ENHANCED. EXTENDS MODELING TECHNIQUES. ADVANCED IMAGE ANALYSIS AND GRAPHICS. BETTER SCIENCE OVERALL.
CODE 700:	ON-BOARD PROCESSING AND DATA COMPRESSION TECHNIQUES FOR ADVANCED INSTRUMENT DESIGN AND ROBOTIC INSTRUMENT SERVICING
CODE 500:	REQUIRES HIGHLY FOCUSED, SHORTER TERM RESEARCH IN HIERARCHIAL EXPERT SYSTEMS, KNOWLEDGE-BASED DEVELOPMENT, MACHINE LEARNING, DATA STRUCTURES, ETC. WOULD PROVIDE NEEDED UNIVERSITY SUPPORT TO PROGRAM NEEDS
CODE 200:	POSSIBLE HETEROGENEOUS DATA BASES, DATA STORAGE AND MANIPULATION TECHNIQUES. ADVANCED NETWORKING.
CODE 400:	ADVANCED FLIGHT SOFTWARE TECHNIQUES
CENTER:	REPUTATION AND STATURE IN SCIENCE COMMUNITY

SUMMARY

- THE CENTER OF EXCELLENCE PROVIDES NASA/GSFC WITH DIRECT ACCESS TO THE ONLY COMMUNITY THAT CAN SOLVE THE OSSA DATA PROBLEMS

- GODDARD IS PREPARED TO DEVOTE THE IN-HOUSE RESOURCES NECESSARY TO MANAGE AND SUPPORT THE CENTER OF EXCELLENCE

- CONCEPT ESTABLISHES A MAJOR NEW RESEARCH APPROACH TO UNIVERSITY AND INDUSTRIAL RELATIONS;

- DEVELOPS MECHANISM TO ATTRACT HIGHLY-TRAINED COMPUTER SCIENTISTS

APPENDIX A

MEMORANDUM OF UNDERSTANDING

between

The University of Maryland

and

The Universities Space Research Association

Purpose: This Memorandum of Understanding summarizes the points of agreement between the University of Maryland and the Universities Space Research Association (USRA) regarding the proposed Center of Excellence in Space Data and Information Sciences at NASA's Goddard Space Flight Center (GSFC).

Points of Agreement:

1. The search committee for the Director shall be a subcommittee of the planning committee for the Center of Excellence, and one of the representatives of the University of Maryland on the planning committee shall chair the search committee. In addition, a member of the Department of Computer Science will serve on the search committee and coordinate the search and selection activity with departmental review, including all normal university procedures.

2. The selection of the Director of the Center of Excellence must be approved by the University of Maryland, USRA, and NASA. It is expected that the selected individual will serve as Director for at least five years. The University of Maryland will provide a line in its budget for the Director, with academic year salary for the line.

3. The Director will be a tenured faculty member in the Computer Science Department at the University of Maryland. To serve as Director of the Center of Excellence, the selected individual will be given 50% release from the normal teaching load.

4. Fifty percent of the Director's academic year salary, all of the summer salary, and the pro-rata associated fringe benefits that are normally available to faculty members at the University of Maryland shall be reimbursed by USRA to the University of Maryland as compensation for the Director's responsibilities to the Center of Excellence.

5. If the selected Director requires reimbursement for the expense of relocating to the Washington area, USRA will bear the cost of the Director's moving expenses.

6. Annual salary adjustments will be negotiable between the Director and the University of Maryland. USRA will honor reasonable adjustments established by these negotiations.

7. The Department of Computer Science will further participate and support the research goals of the Center of Excellence through a contract between USRA and the University, with the Chairman of the Department of Computer Science serving as the manager of the contract. In addition to USRA's share of the salaries of the Director, half-time academic year support and full-time summer support will be provided for three University of Maryland, College Park, faculty members and three graduate students. These faculty members shall be designated as Staff Scientists in the Center of Excellence and shall hold joint appointments with the Center of Excellence and the appropriate academic department at the university. All joint appointments shall be made with the concurrence of the Director, the Department Chairman, and the Provost of the Division of Mathematics, Physical Sciences, and Engineering.

8. USRA's ability to reimburse the University of Maryland for a portion of the salary and fringe benefits of the Director is contingent upon continued funding for the Center of Excellence by NASA. USRA will exert best efforts to negotiate a contract with NASA that provides at least four and one half months (one academic year at half-time) of salary and fringe benefits for the Director in the event that NASA prematurely terminates the contract.

9. The specific terms of this agreement will be reviewed annually and may be modified by mutual written consent of the signing parties.

The University of Maryland

John Toll

John S. Toll
President

The Universities Space
Research Association

Paul J. Coleman, Jr.
President

CONCURRENT PROCESSING ALGORITHM DESIGN
(CPAD)

LOAD BALANCING
IN A PARALLEL PROCESSING ENVIRONMENT

MOKTAR SALAMA
JPL

NASA COMPUTER SCIENCES & DATA SYSTEMS WORKSHOP
NOVEMBER, 1986
WILLIAMSBURG, VIRGINA

LOAD BALANCING
IN A PARALLEL PROCESSING ENVIRONMENT

What is Load Balancing?

- Parallel Processing: resources of several processors are brought together to share in solution of a single problem.
- Total solution is decomposed into "processes" that execute concurrently and communicate sparingly.
- If the processes require equal resources when run on an ensemble on n-processor ⇒ load-balanced computer.
- If not, then: speed of the ensemble of n-processors is that of the slowest processor (most heavily loaded) ⇒ load-imbalanced computer.

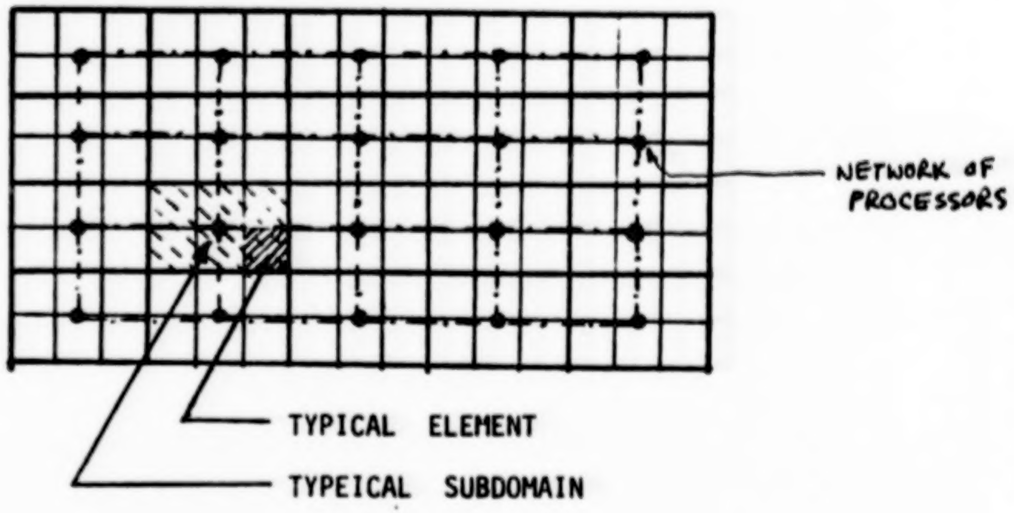
An optimal assignment problem

Factors Affecting Load Balancing

- Amenability of processes to uniform distribution (time to execute, precedence constraints, communication requirements, . . . etc).
- Topology of the ensemble (how many processors, communication topology, communication mechanism, . . . etc).

AN EXAMPLE OF LOAD BALANCING

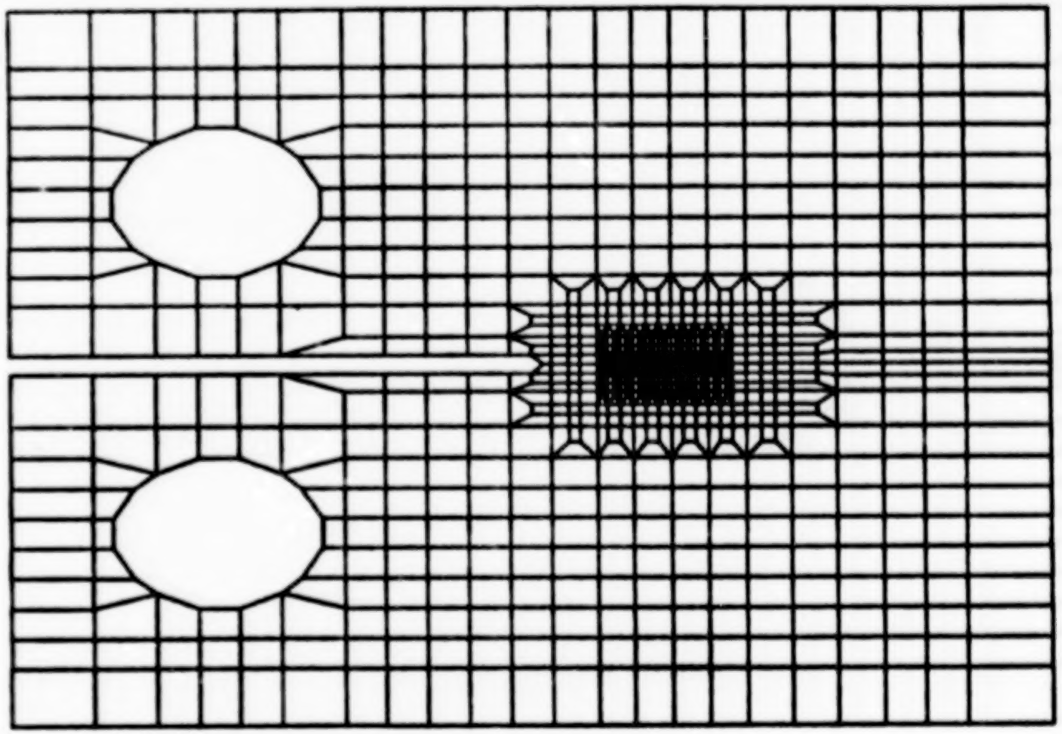
- o Regular domain, regular processor topology


NETWORK OF PROCESSORS
TYPICAL ELEMENT
TYPEICAL SUBDOMAIN

- o Process = subdomain = task
- o Equal task distribution and communication; work load is blananced.

IMBALANCED PROBLEM

O IRREGULAR DOMAIN



- How to map the domain to processors is not obvious.
- SPEED of the ensemble of n-processors is that of the slowest (most loaded) processor.
- Needed; a methodology to balance load distribution (Computation + Communication).
- An intractable problem in combinatorial optimization; hence - heuristic.

THE SIMULATED ANNEALING ANALOGY

A HEURISTIC LOAD BALANCING APPROACH

- In a body of matter, system's energy

  E = {(Repulsive Forces) + (Attractive Forces)}

  In a parallel computer, comp. cost

  $$E = \{(\text{Comp. Cost}) + (\text{Comm. Cost})\}$$
  $$= \left\{ \sum_{i=1}^{N} (W_i)^2 + \frac{t_{comm.}}{t_{comp.}} \sum_{p,q}^{p<q} C_{pq} \right\}$$

  W_i = Workload of processor, i

  C_{pq} = Comm. cost between elements p and q

  $t_{comm.}$ = Typical comm. time per element

  $t_{comp.}$ = Typical time for floating pt. oper/element

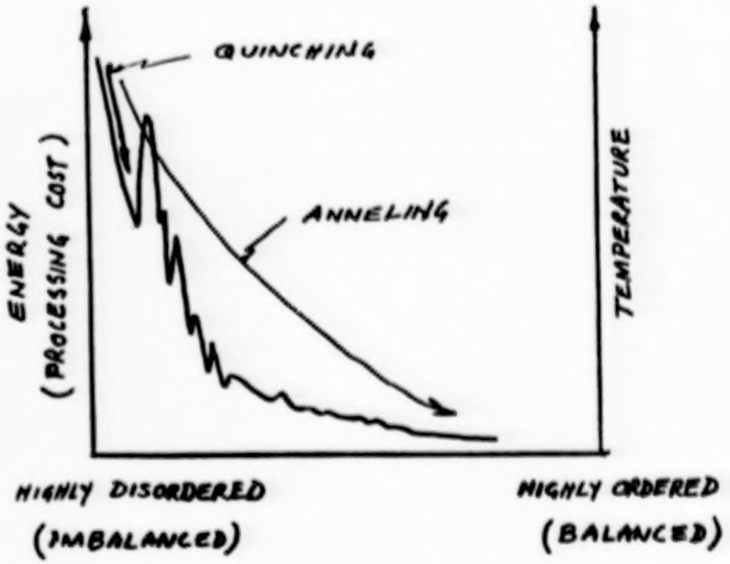
- Positions assumed by atoms in a body determine the body's energy state. Allocating elements to processors determine total processing cost.

SIMULATED ANNEALING ANALOGY (CONT.)

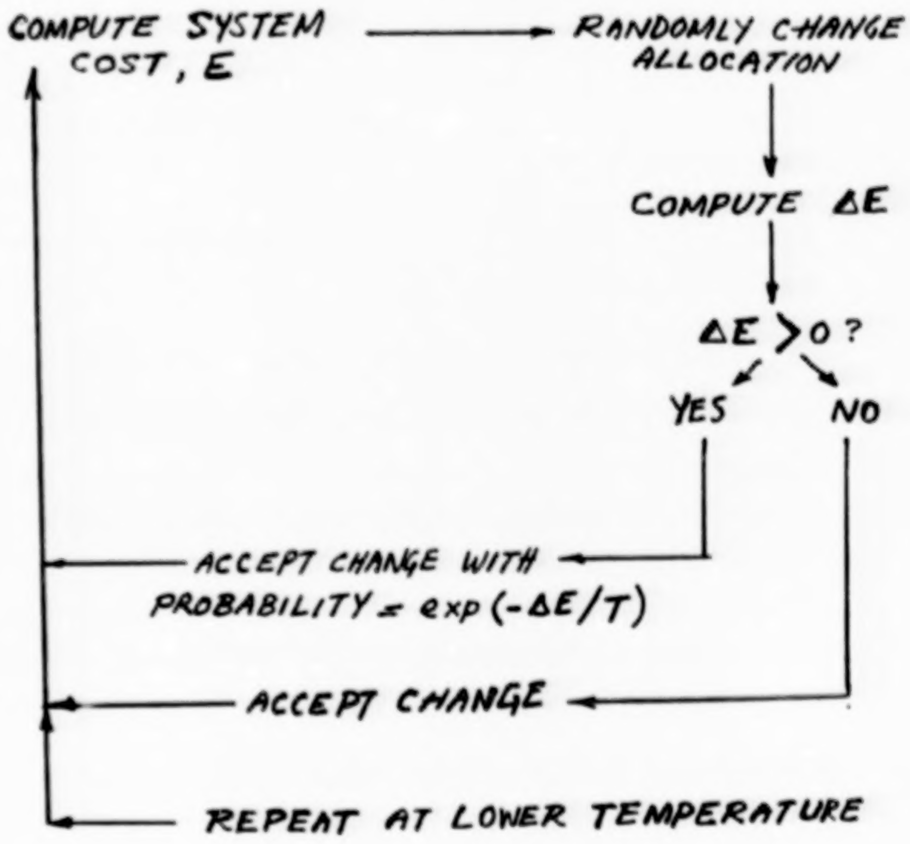
- o At high temp. T, atoms are at random positions (Disordered State) ⇒ High Energy.
  Arbitrary allocation of elements to processors (Load Imbalance) ⇒ High Cost.

- o Cooling from T→o, allows atoms to migrate to an Ordered State of Low Energy. Degree of order depends on cooling rate:
  Slow cooling (annealing) lower ground state than rapid cooling (quinching).

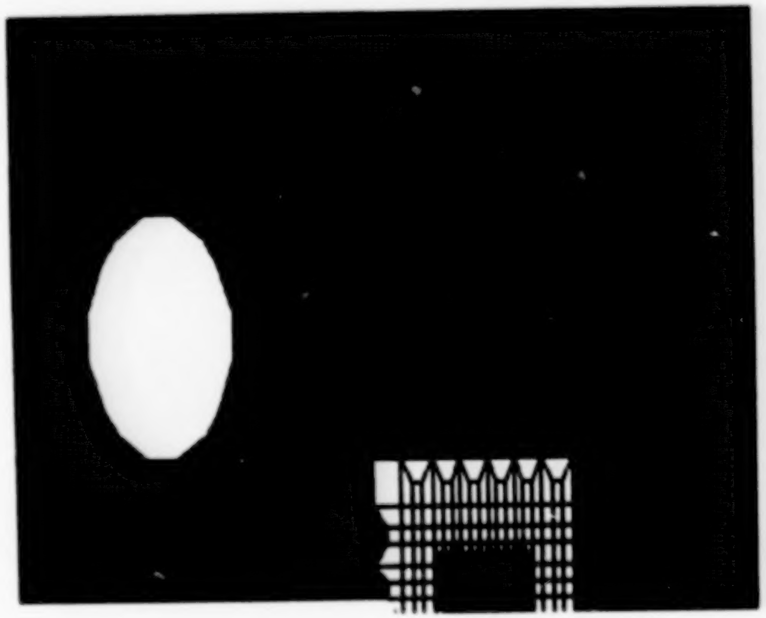
As initially assigned elements are allowed to migrate to processors (using monte carlo - based moves), The system becomes balanced.



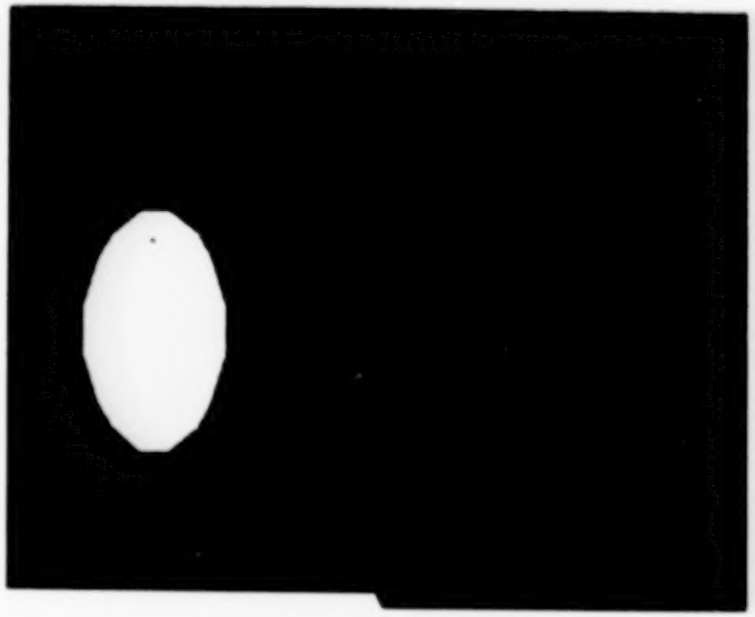
- ALGORITHM


COMPUTE SYSTEM COST, E
RANDOMLY CHANGE ALLOCATION
COMPUTE ΔE
ΔE > 0 ?
YES
NO
ACCEPT CHANGE WITH PROBABILITY = exp (-ΔE/T)
ACCEPT CHANGE
REPEAT AT LOWER TEMPERATURE

ORIGINAL PAGE IS
OF POOR QUALITY

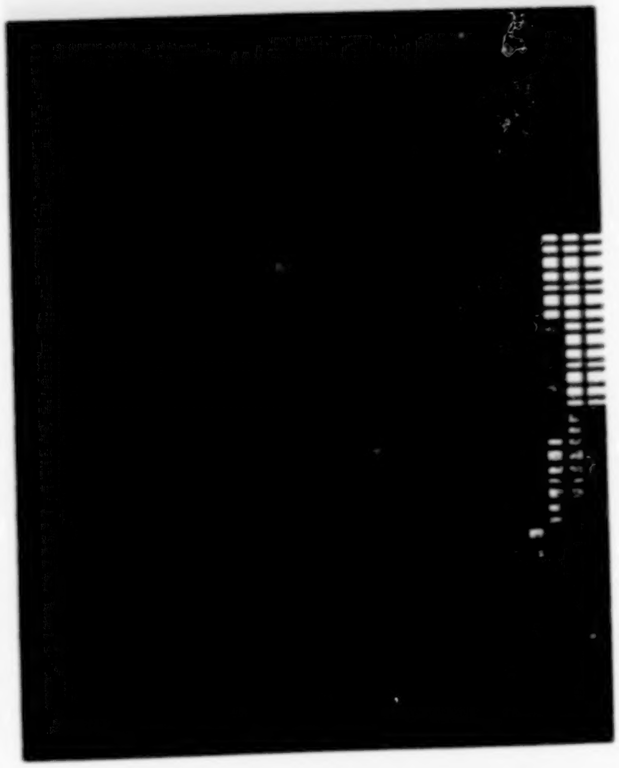


INITIAL ASSIGNMENT: WORKLOAD VARIATION 6 TO 331



ASSIGNMENT AFTER 20 CYCLES: WORKLOAD VARIATION 33 TO 35

ORIGINAL PAGE IS
OF POOR QUALITY



DETAILS OF THE INTENSELY PACKED REGION

FUTURE PLAN

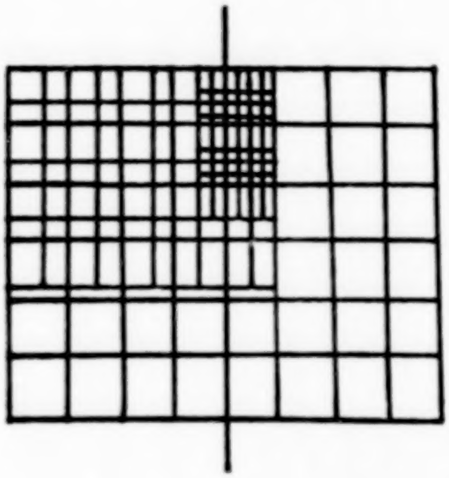
- Load Balancing in MAX's Data Flow Environment
    - o Static load balancing: assignment of processes to processors is done prior to execution and never altered.
    - o Dynamic load balancing: assignment of processes to processors is done dynamically as execution progresses.
- Develop Other Heuristics

Performance Critical Decisions in Parallel Scientific Computations

David M. Nicol

*Institute for Computer Applications
in Science and Engineering*

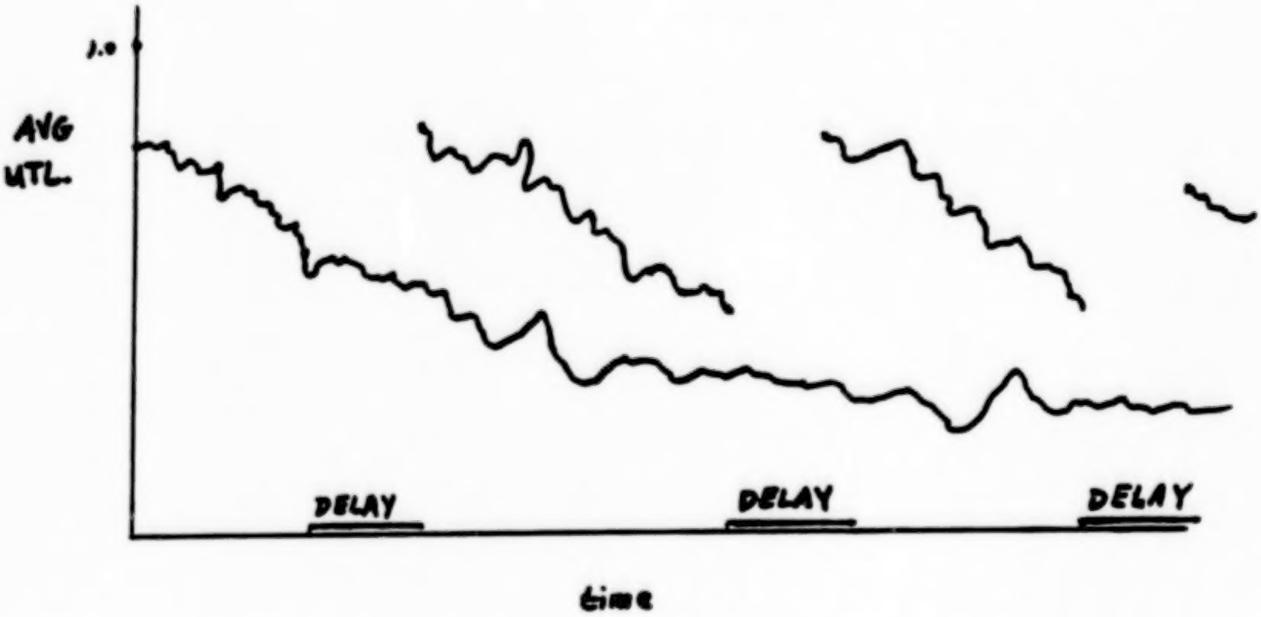
Mapping of parallel scientific computation to message passing architectures is usually domain oriented



- Workload can change dynamically
  - When should we remap?
- Convergence testing performed every iteration
  - How can we avoid excessive overhead?

Workload dynamics: continuous change

- e.g. Vortex methods for incompressible fluid flow



Optimal Remapping Decision Policy

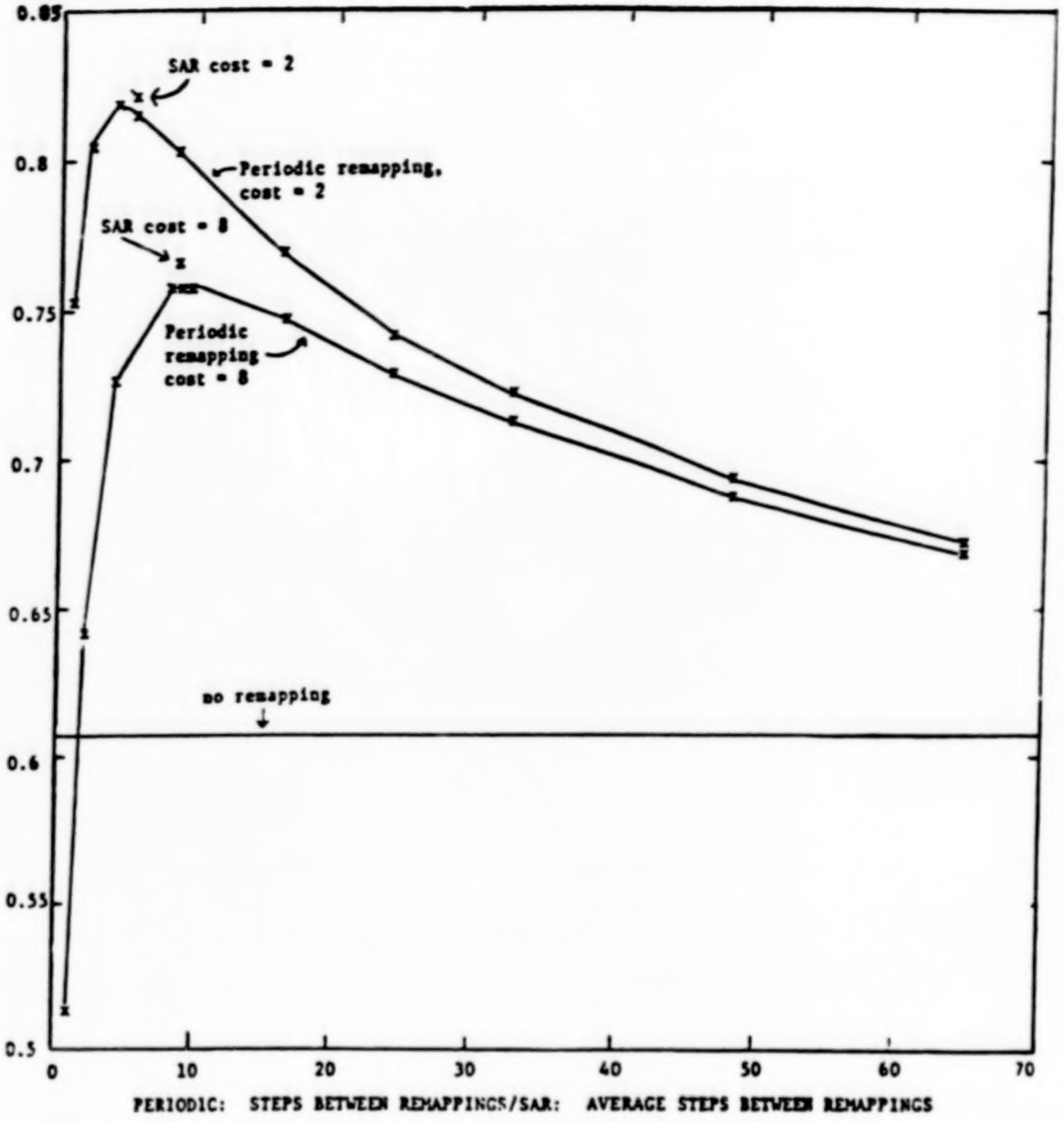
- Calculation of MDP possible with small model
- Optimal policy knows "too much" about system

Decision Heuristic

- Attempt to minimize average cost per step

$$W(n) = \frac{\sum_{i=1}^{n} (max(i) - avg(i)) + C}{n}$$

- As function of n $E[W(n)]$ has at most one local minimum
- How to tell when $W(n)$ minimized?
    - — Approximate with SAR

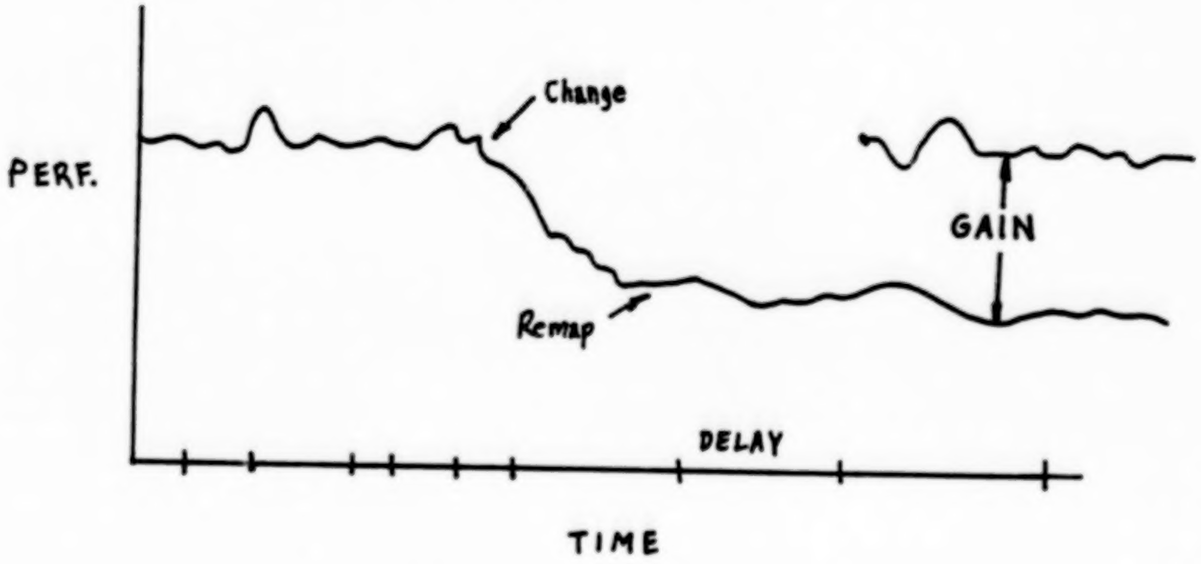


MUM Model: Performance of SAR Compared with Performance of Periodic Remapping.

Eight chains, 400 steps, each chain has 19 states, p = 0.5, each data point calculated from 200 sample paths.

Workload dynamics: abrupt change

- e.g. Multi-phase computations



Remapping Decision Heuristic

- Many decision model parameters difficult to estimate
- Examined fixed ρ-threshold policy
- ρ ∈ [.7,.8] nearly optimal within envelope of admissible costs and gains
  - Difference from NR policy depends on gain
  - Keep ρ high to avoid unneeded overhead, low to avoid inaccurate test statistics
  - Policy tolerates mis-estimation of statistical parameters
- Conclusion: Remapping decision is manageable, focus on determining when gain is possible
  - in envelope?
  - gain potential?

Convergence Checking

- Goal is to reduce overhead by reducing number of checks
- Need to balance cost of check with threat of overshoot
- Choose iteration (offset) n to maximize average number of "useful" iterations

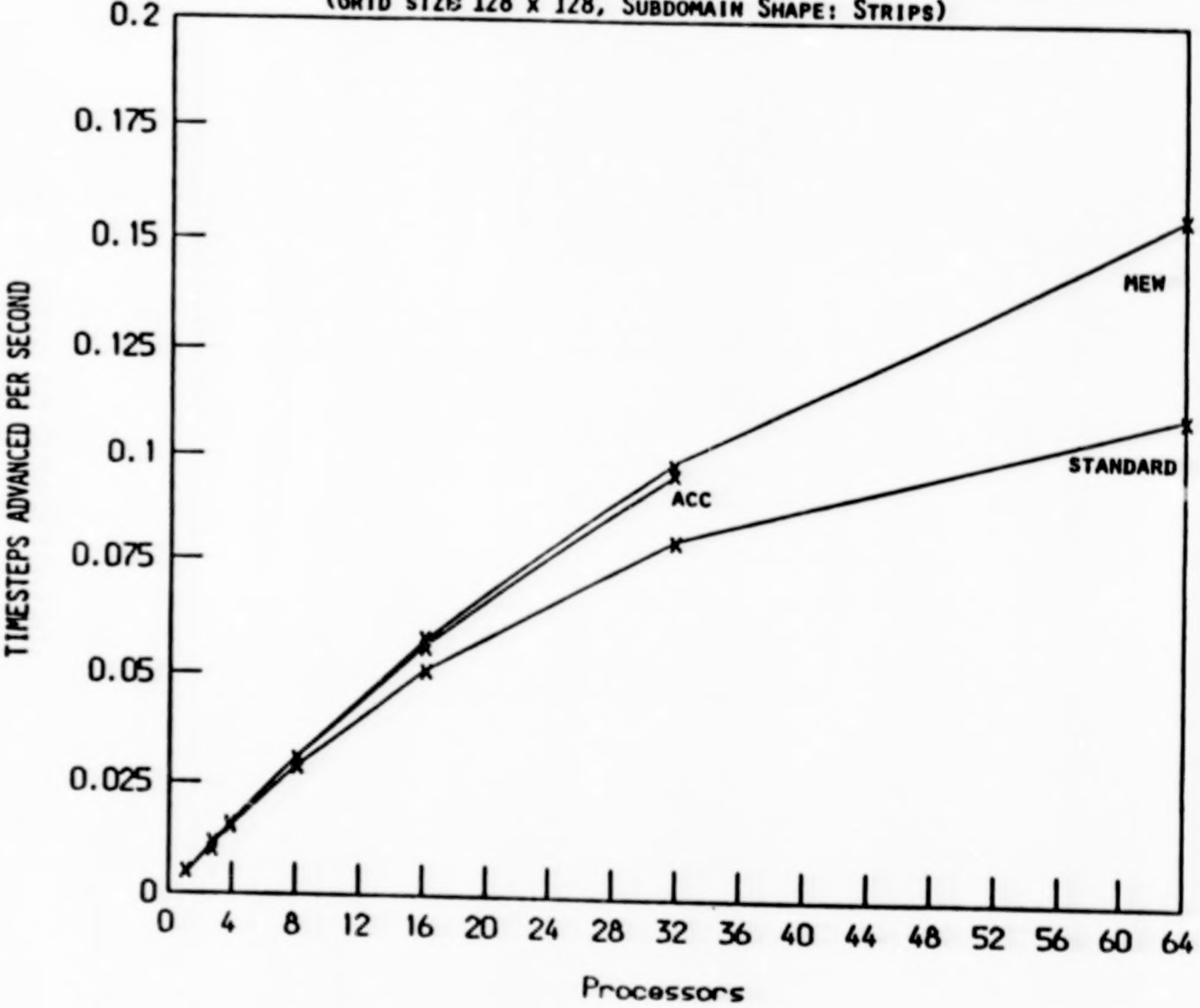
$$\frac{\sum_{i=1}^{n} Prob\{\text{Iteration } i \text{ is needed}\}}{n \cdot D + I}$$

  - D iteration cost
  - I cost of convergence check

- Probability terms computed using Bayesian techniques

EFFECT OF CONVERGENCE CHECKING SCHEMES ON ALGORITHM PERFORMANCE.

(GRID SIZE 128 x 128, SUBDOMAIN SHAPE: STRIPS)



Conclusions

- High performance in parallel scientific computations requires treatment of decision problems
  - Remapping decisions
  - Convergence checking decisions
- Simple "greedy" heuristics work nearly optimally
- Need to ensure that decision policies can be implemented at system level

Sparse Distributed Memory

M. Raugh/RIACS

This work entails:

1. Programming a simulator of the proposed architecture on the Intel iPSC parallel computer or the Symbolics Lisp machine or both.

2. Developing simple models of an environment and of sensory input to and effector output from memory and use the simulator to study learning by trial and error in this simple environment.

3. Comparing the models developed in (b) to human and animal sensory and effector systems operating in the real world and prepare recommendations about further research.

4. Preparing recommendations about the building of special-purpose hardware for sparse, distributed memories and about the uses of such memories.

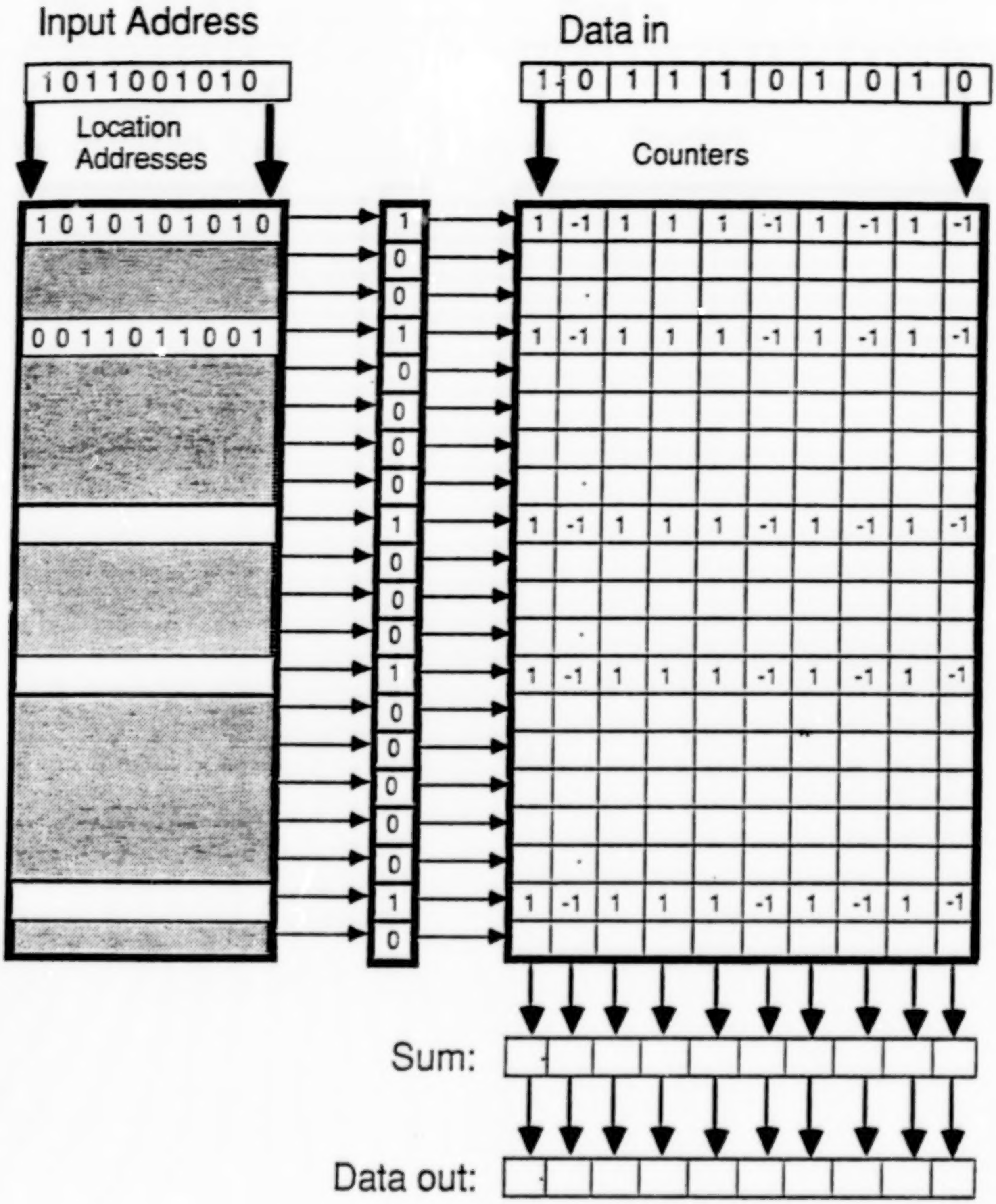
5. Relating findings of this research to facts about human memory and perception and prepare recommendations for the design of artificial environments for humans, with emphasis on environments where the cost of human error is very high.

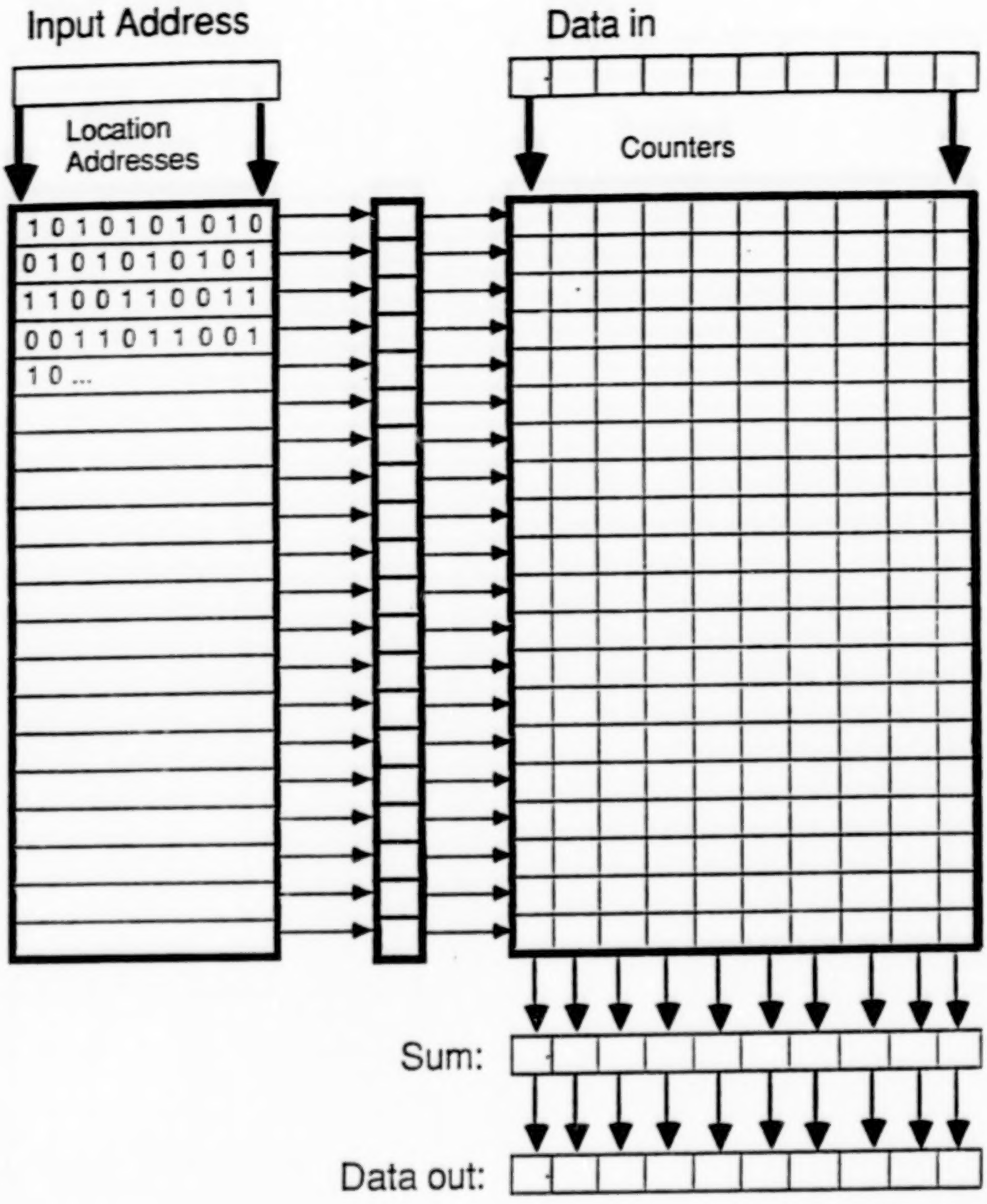
Scope of Task

- Feasibility study
    - Assess merits of SDM
    - Decide whether and how to build
        - SDMs for real tasks
        - Full–scale VLSI implementation
- A fundamental question
    - How to code the sensory inputs?
    - For research applications in
        - Robotics
        - Pattern recognition
        - Cognitive science
        - Interpretation of continuous speech

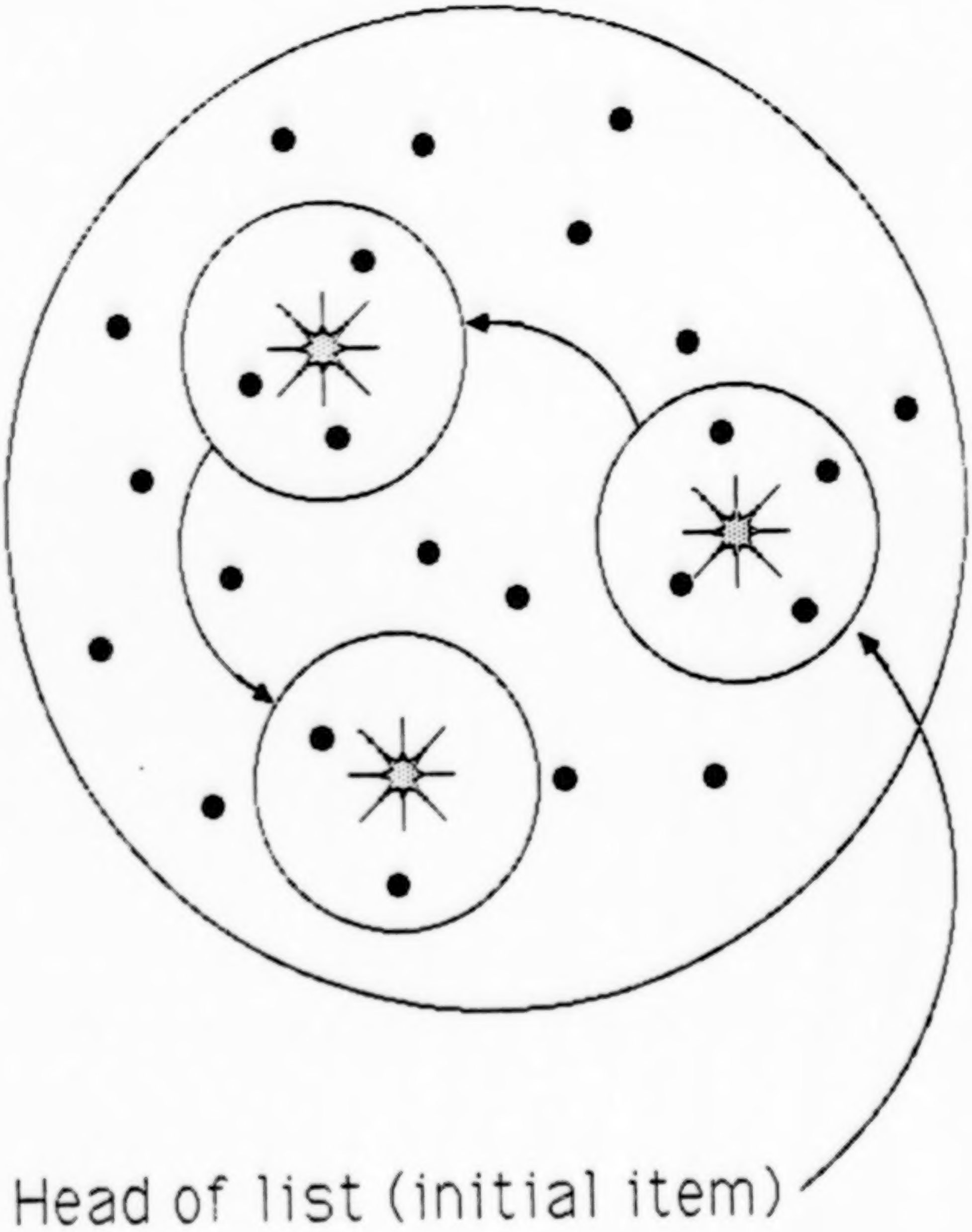
Theory Yields New Computer Architecture

- Distinction between memory and processing disappears
- Processing
  - Massively parallel
  - Distributed throughout memory
- 10–100 memory locations on a chip
- A computer based on the SDM

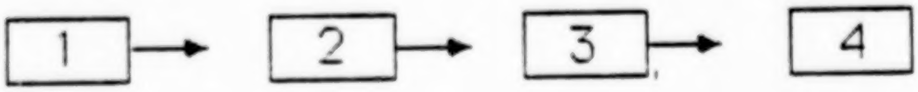

Input Address
1011001010
Location Addresses
Data in
1 0 1 1 1 0 1 0 1 0
Counters
1010101010
0011011001
1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0
1 -1 1 1 1 -1 1 -1 1 -1
Sum:
Data out:


Input Address
Data in
Location Addresses
Counters
1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 1
1 1 0 0 1 1 0 0 1 1
0 0 1 1 0 1 1 0 0 1
1 0 ...
Sum:
Data out:

Items may be stored as a linked list



Head of list (initial item)



Main Properties of Model Memory

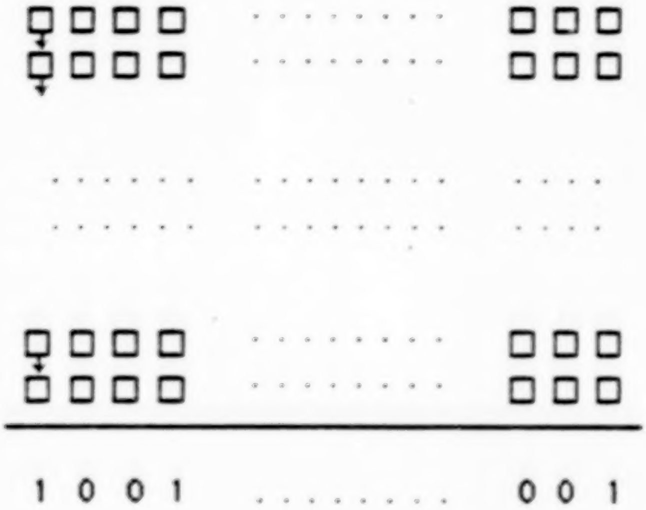
- Addresses can be chained (linked list)
  - Stream of recall
  - Causal relations
- SDM is sensitive to similarity
  - If you "know" 600 bits
    - And correctly "guess" 200 more
      - You will probably converge to correct memory
- SDM is capable of forgetting!
  - But can be reinforced by rehearsal

```
 Another initial condition? (1-yes)
1
 random (0) or perturbed initial condition(1)?
1
 Which memory state to perturb around?
6
 With what noise level? (0.0 to 1.0)
.2
 The initial condition is:
    XX  XXXX          XX
XX        XXXX        XX
  XXXX    XX      XXXXXXXXXXXX
  XXXXXX  XXXXXXXXXXXXXX  XXXX
  XXXX    XXXXXXXXXXXX      XX
  XXXX    XXXX  XXXXXX      XXXX
XXXX    XXXXXX    XXXXXXXXXXXX
      XXXX  XX    XXXXXX  XX
XXXXXX  XXXXXXXX  XXXX  XX
XX  XX  XXXX    XXXXXXXX    XX
  XXXX    XX        XX      XX
  XXXXXXXXXX  XX  XXXXXX
  XXXX            XXXX  XX  XX
  XX      XX      XXXX
    XXXXXXXXXX  XXXXXXXX    XX
  XX  XX
 (initial condition)
 type 1 ‹or› to continue
1
 selected locations-11
        XXXX

  XXXX    XXXX    XXXXXXXXXX
  XXXX    XXXXXX  XXXXXXXXXXXX
  XXXX    XXXXXX  XXXX      XX
  XXXX    XXXX  XXXXXX    XXXX
  XXXX    XXXX    XXXXXXXXXXXX
  XXXXXXXXXXXX    XXXXXXXXXX
  XXXXXXXXXXXXXX  XXXX
  XXXX    XXXX    XXXX
  XXXX    XXXX      XX
  XXXX    XXXXXX  XXXX
  XXXX    XXXX    XXXX
  XX      XXXX    XXXX
```

```
 Another iteration with this initial condition (1)?
1
 selected locations-13

  XXXX    XXXX    XXXXXXXXXX
  XXXX    XXXX    XXXXXXXXXXXX
  XXXX    XXXX    XXXX    XXXX
  XXXX    XXXX    XXXX    XXXX
  XXXX    XXXX    XXXXXXXXXXXX
  XXXXXXXXXXXX    XXXXXXXXXX
  XXXXXXXXXXXX    XXXX
  XXXX    XXXX    XXXX
  XXXX    XXXX    XXXX
  XXXX    XXXX    XXXX
  XXXX    XXXX    XXXX
  XXXX    XXXX    XXXX
  XXXX    XXXX  XXXXXXXX

 Another iteration with this initial condition (1)?
1
 selected locations-14

  XXXX    XXXX    XXXXXXXXXX
  XXXX    XXXX    XXXXXXXXXXXX
  XXXX    XXXX    XXXX    XXXX
  XXXX    XXXX    XXXX    XXXX
  XXXX    XXXX    XXXXXXXXXXXX
  XXXXXXXXXXXX    XXXXXXXXXX
  XXXXXXXXXXXX    XXXX
  XXXX    XXXX    XXXX
  XXXX    XXXX    XXXX
  XXXX    XXXX    XXXX
  XXXX    XXXX    XXXX
  XXXX    XXXX    XXXX
```

Reading is columnar addition:

Add corresponding counters from all selected locations



Each 0-bit results from a negative column-sum

Each 1-bit results from a positive column-sum

Writing:

Each physical memory location contains 1,000 counters

□ □ □ □ · · · · · · · · · □ □ □

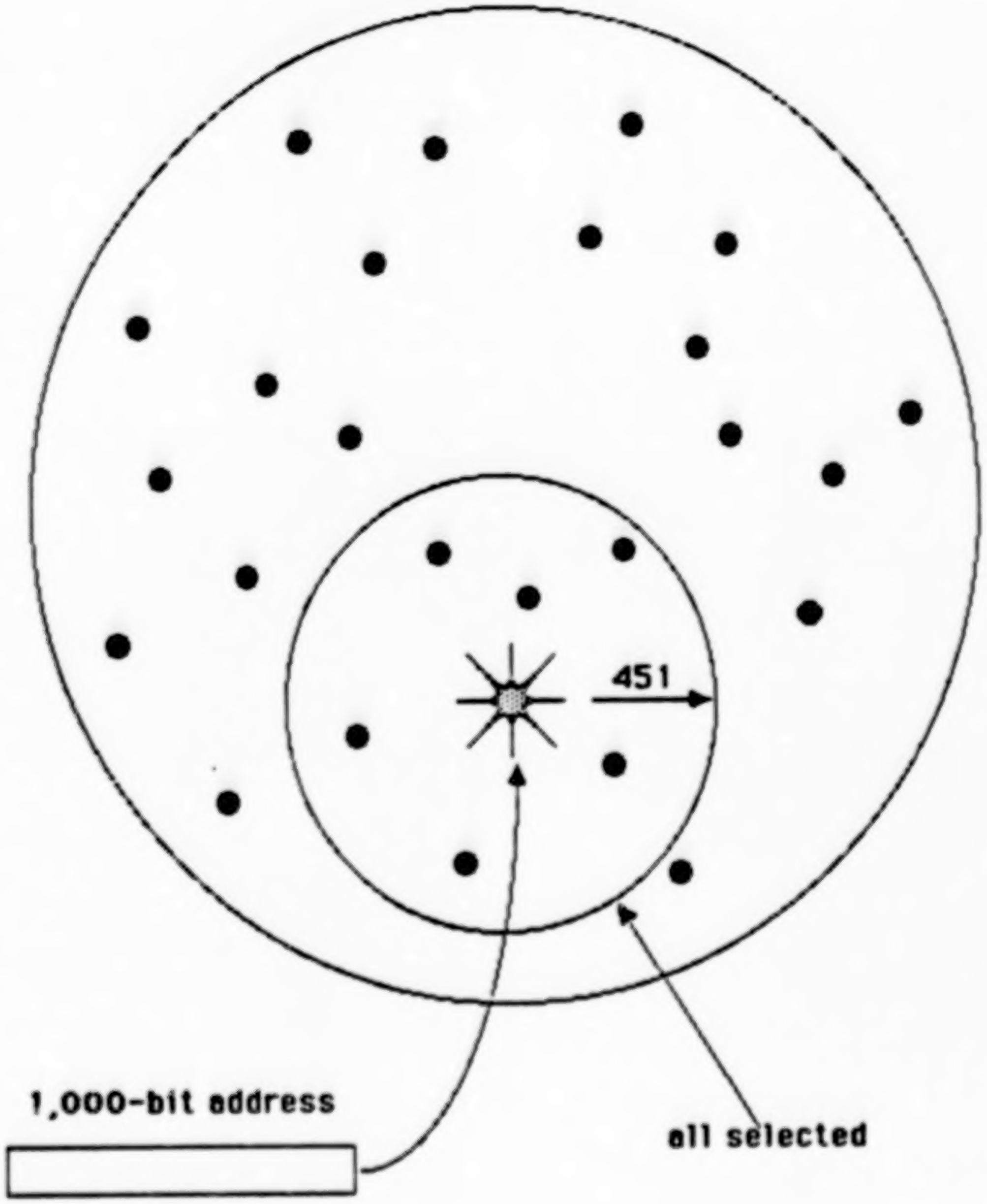
Each address contains 1,000 bits

1 0 0 1 . . . . 0 0 1

Each 1-bit adds 1 to the corresponding counter

Each 0-bit subtracts 1 from the corresponding counter

Addressing

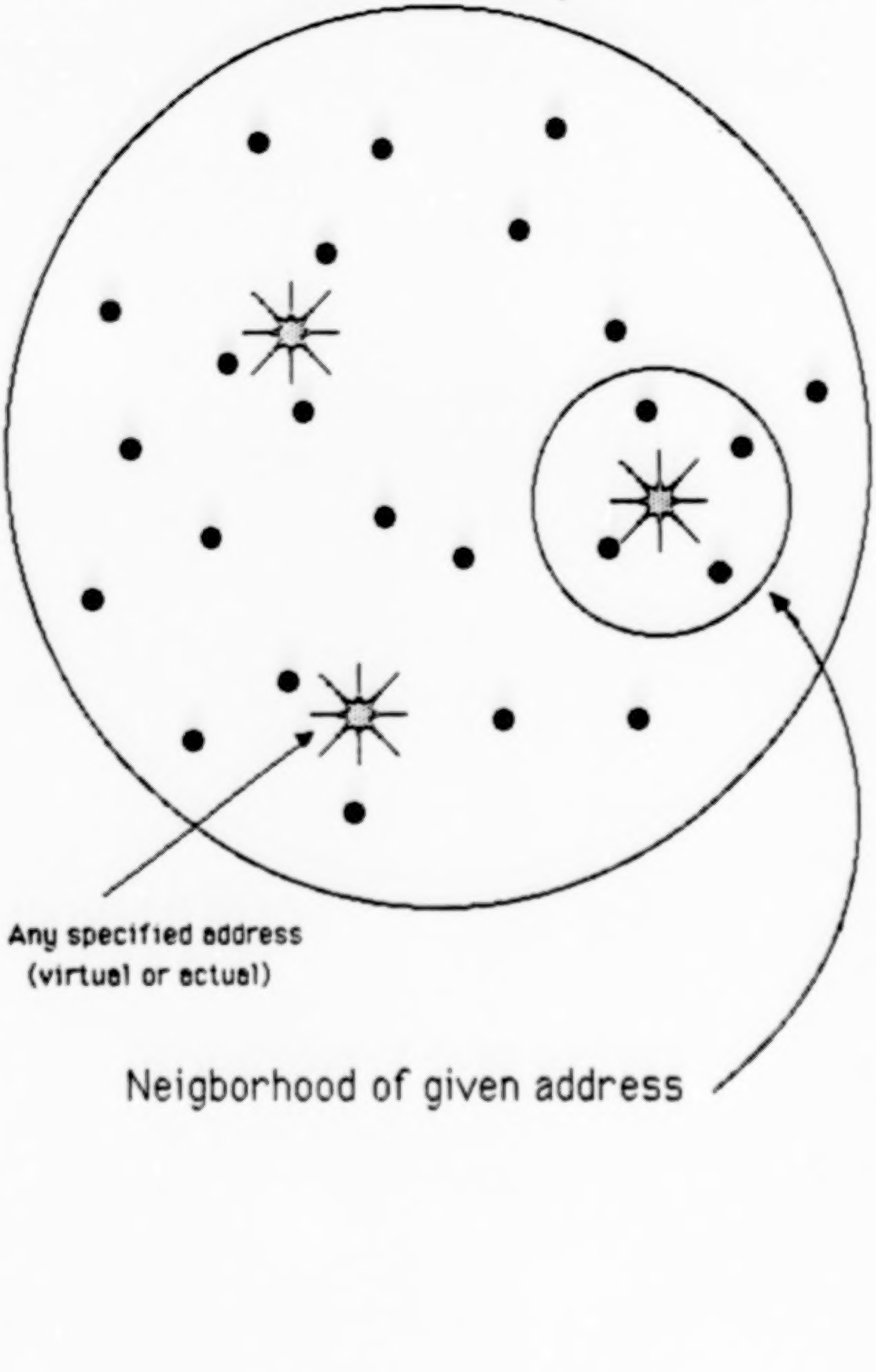

451
1,000-bit address
all selected

Gist of Theory (II)

- Sparse, ***distributed*** memory
- Reading and writing at specified address
- (Hamming distance between binary memory addresses)
  - Write to locations within radius of 451
    - Yields on average 1,000 physical locations
  - Read from locations within radius of 451

Huge Virtual Address Space

- Few actual memory locations



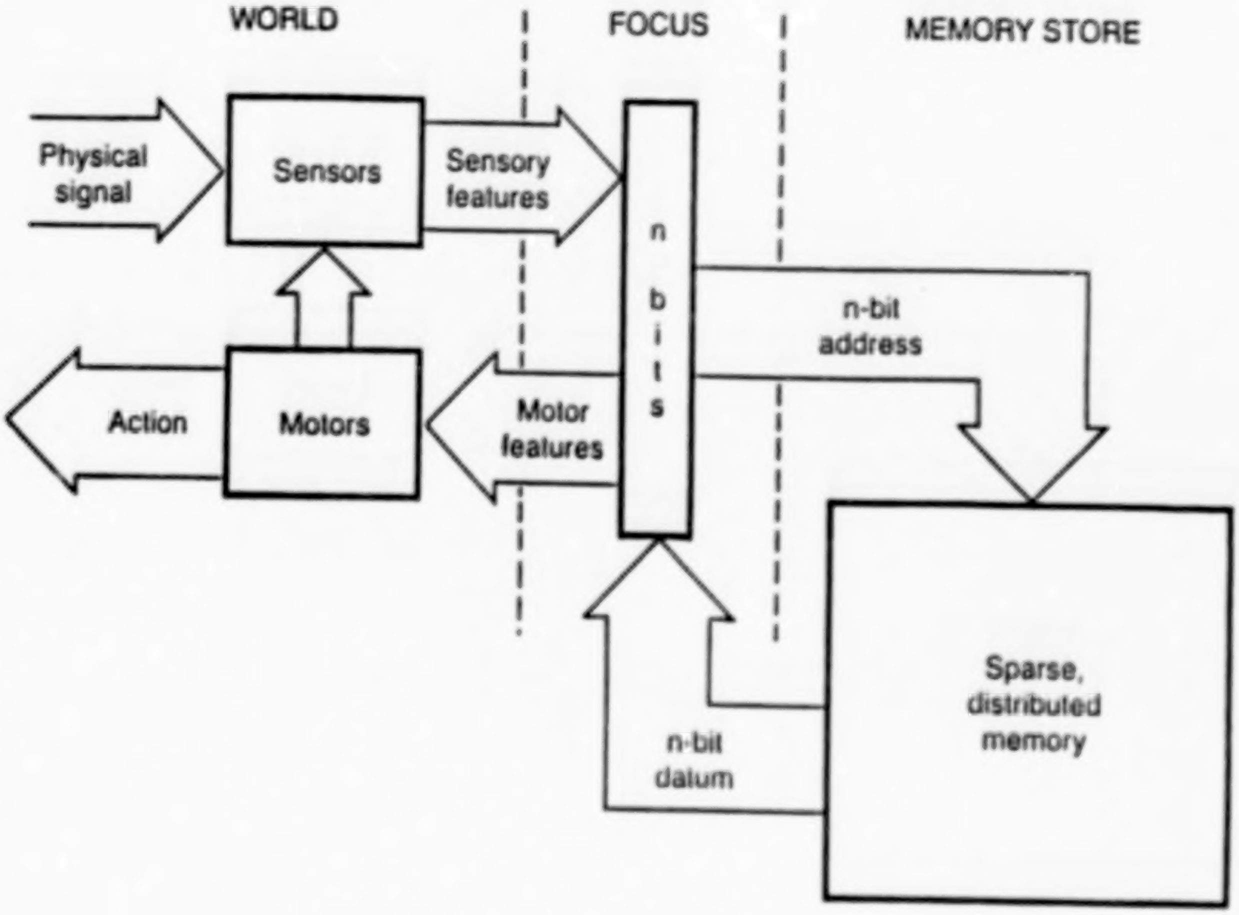
Any specified address
(virtual or actual)
Neigborhood of given address

Gist of Theory (I)

- ***sparse,*** distributed memory
    - Huge (virtual) address space: 2^{1000}
    - Few (actual) physical memory locations: 2^{20}
        - Sprinkled randomly throughout the address space

Preliminary Assumption

- 1,000 bits code one moment of experience
    - Each coded moment serves as an address
- Kanerva postulates the existence of a *focus*
    - Clearing point for data going into or out of memory
- The coding problem is not addressed by Kanerva's existing theory
    - but will be addressed by this task


WORLD
FOCUS
MEMORY STORE
Physical signal
Sensors
Sensory features
n bits
n-bit address
Action
Motors
Motor features
Sparse, distributed memory
n-bit datum

Sparse, Distributed Memory

- Massively parallel architecture
- Information stored in large number of neighboring addresses determined by "content"
- Memory is dynamic:
    - changes with experience
        - learns
            - causal relationships
            - time–varying phenomena
- Memory is sensitive to similarity
    - and it is
        - forgetful !!!
        - (but can be reinforced)

Kanerva's Theory of Sparse, Distributed Memory (SDM)

- A model of human memory
  - Explicit
  - Simple
  - Engineerable
- Dissertation being published by Bradford Books of MIT Press

PROJECT: SPARSE DISTRIBUTED MEMORY

CHARACTERISTICS:

- Highly–parallel pattern computer
- Based on Kanerva's mathematical theory of human memory

PARTICIPANTS:

- Pentti Kanerva (RIACS), Project Manager
- Harrison Leong (RIACS)
- David Nagel (Ames)
- Michael Flynn (Stanford)

STATUS:

- Completed simulators
- Building digital prototype

DISTRIBUTED OPERATING SYSTEMS
FACT OR FANTASY?

ED FOUDRIAT
NASA - LaRC

- DOS/OS COME IN MANY FLAVORS
- ADDRESS LAN WITH MID TO HIGH COMMUNICATIONS ≈ 10^7 B/S

SHARED MEMORY	VS	NON SHARED MEMORY
PARALLEL	VS	DISTRIBUTED
ADDRESS OP FETCH/STORE	VS	MESSAGE OP SEND/RECEIVE
COMMUNICATIONS UNDER O/S	VS	COMMUNICATIONS IN O/S

DOS LIKE PARALLEL SUPPORTS DISTRIBUTED PROCESSING, DATA, & CONTROL

QUESTION - UNDER WHAT CONDITIONS IS EACH APPLICABLE ?

ORIGINAL PAGE IS
OF POOR QUALITY

A CASE FOR SHARED MEMORY

MESSAGE STEPS :

1. APPL. PROCESS PREPARE INFO - CALL SEND
2. SEND PROCESS - PREPARE HEADER & CONTROL - SEND
3. RECEIVE MESSAGE - HANDLE CONTROL - LOCATE RECIPIENT
4. RECIPIENT PROCESSING.

OPERATIONAL STEPS:

1 CONTEXT SWITCH
2 COPY
3 LOCATE - ROUTING OPS
4 COMM. PROTOCOL OPS - SEQ, FLOW

5. PHYSICAL TRANSMISSION

6 HANDLE RECEIVE INTERRUPT
7 SET COMMUNICATION STATE
8 LOCATE RECIPIENT
9 COPY
10 CONTEXT SWITCH

SUPPOSE EACH STEP .1 - 1.0 MSEC

SUPPOSE REDUCE STEPS TO 3-5

MESSAGES MINIMUM 1MSEC - 10MSEC.

OBSERVATIONS

CPU OPERATIONS ARE LIMITING FACTOR

COMM. IMPROVEMENTS - 10 - 10^3
CPU " - 10

BEST MESSAGE SYSTEMS .5 MSEC.

SUPPORT:

ED GERINGHER - N.C. STATE E & CE DEPT.

CM* MESSAGE SYSTEM 14X SLOWER

PARALLEL MACHINES USE FETCH/STORE

ORIGINAL PAGE IS
OF POOR QUALITY

Conclusion

For LAN with Mid Communications Rates:

- All Basic Comm. Established Backplane to Backplane in Hardware
- Serial CPU Activity in Comm. Not Practical
- Nonshared Memory DOS for Autonomous and/or Low Comm Rate Environments - NOS
- Asynchronous Message as Basic (Low Level) DOS Paradigm Not Practical

High Performance Based DOS is a Fantasy.

When Performance Counts Don't Send Messages.

Parallel / Shared Memory Systems

- Crossbar other Switched Memory
- Partitioned - Local Cache with Controls
- Segregated Memory Blocks
- Neighborhood Connectivity Cubes, Meshes, etc

Basic Problems

Naming that's Hardware Compatible

Consistency Under Parallel Ops

Maintaining Correlation over Operation
(Latency)

Maintaining Operations Under
Failure

Two Basic Memory Operations

- Fetch / Store
  - Asynchronous Op
  - Hardware Resolved
  - Duplicate / "Multicate" Information
  - Short Time Frame Op.
- Send / Receive (Message) Connect (Virtual Circuit)
  - Synchronous Op.
  - Hardware Resolve . CPU Controlled
  - Duplicate / "Multicate" Information
  - Long Time Frame Op

Note : Full Duplex Virtual Circuit Satisfies Sequencing and Failure Requirements. Other Distributed/Parallel Problems Have High Commonality

CONCLUDING REMARKS

- HIGH PERFORMANCE DISTRIBUTED AND PARALLEL COMPUTING SYSTEMS ARE LOGICAL EQUIVALENTS
  - SDI, SPACE STATION, ADVANCED EMBEDDED WILL USE SHARED MEMORY
- DIST. "SHARED MEMORY" HARDWARE CONCEPTS NEED IDENTIFICATION & SELECTION
- A COMBINED FETCH/STORE, CONNECT MACHINE STUDY NEEDED
- RESEARCH ON MESSAGE PROTOCOL OPERATIONS, ETC. CAREFULLY REDIRECTED
- CLOSER ALLIANCE BETWEEN DISTRIBUTED & PARALLEL RESEARCH ACTIVITIES -- LOTS OF COMMONALITY

SPACE & EARTH SCIENCES DIRECTORATE

CODE 600

DISTRIBUTED ACCESS VIEW INTEGRATED DATABASE (DAVID) SYSTEM

HETEROGENEOUS DISTRIBUTED DATABASE MANAGEMENT-- THE DAVID SYSTEM

Barry E. Jacobs
Senior Research Scientist
NASA/GSFC

NOV. 19, 1986

DRO058 A-2

SPACE & EARTH SCIENCES DIRECTORATE
CODE 600

DISTRIBUTED ACCESS VIEW INTEGRATED DATABASE (DAVID) SYSTEM

GSFC

PROBLEM:

NASA SCIENTISTS AND MANAGERS HAVE TO LEARN MANY DIFFERENT ACCESS METHODS IN ORDER TO OBTAIN DATA.

DEVICES	OPER SYS	DBMS'S	COMMUNICATIONS	NASA DATA SYSTEMS
IBM	DOS	IMS	ARPA NET	NASA OCEAN DATA SYSTEM (NODS)
VAX	UNIX (BSD)	DB2	DEC NET	NASA LAND DATA SYSTEM (NLDS)
CDC	UNIX (AT&T)	ORACLE	BIT NET	TECHNICAL MANAGEMENT INFORMATION SYSTEM (TMIS)
SUN	VMS	INGRES	DIAL-UP	NASA ASTROPHYSICS DATA SYSTEM (NADS)
CRAY	IBM OS	SEED	IBM NET	
		CODASYL		

[illegible]

SPACE & EARTH SCIENCES DIRECTORATE

CODE 600

DISTRIBUTED ACCESS VIEW INTEGRATED DATABASE (DAVID) SYSTEM

GSFC

SOLUTION:

DISTRIBUTED ACCESS VIEW INTEGRATED
DATABASE (DAVID) SYSTEM

DEVICES	OPER SYS	DBMS'S	COMMUNICATIONS	NASA DATA SYSTEMS
IBM	DOS	IMS	ARPA NET	NASA OCEAN DATA SYSTEM (NODS)
VAX	UNIX (BSD)	DB2	DEC NET	NASA LAND DATA SYSTEM (NLDS)
CDC	UNIX (AT&T)	ORACLE	BIT NET	TECHNICAL MANAGEMENT INFORMATION SYSTEM (TMIS)
SUN	VMS	INGRES	DIAL-UP	NASA ASTROPHYSICS DATA SYSTEM (NADS)
CRAY	IBM OS	SEED	IBM NET	
		CODASYL		

SPACE & EARTH SCIENCES DIRECTORATE

CODE 600

DISTRIBUTED ACCESS VIEW INTEGRATED DATABASE (DAVID) SYSTEM

GSFC

OUTLINE OF TALK

- WHAT IS DAVID?
- USER'S VIEW OF DAVID
- DEMONSTRATION SYSTEM
- ASTROPHYSICS DATA SYSTEM
- HOW DAVID WORKS

DISTRIBUTED ACCESS VIEW INTEGRATED DATABASE (DAVID) SYSTEM

WHAT IS DAVID?

- DAVID IS A SYSTEM FOR ACCESSING HETEROGENEOUS DISTRIBUTED DATABASES.
- DAVID IS A STANDALONE HOMOGENEOUS DISTRIBUTED DATABASE MANAGEMENT SYSTEM.
- DAVID IS A HETEROGENOUS DISTRIBUTED OPERATING SYSTEM.
- DAVID IS A HETEROGENEOUS COMMUNICATIONS SYSTEM.

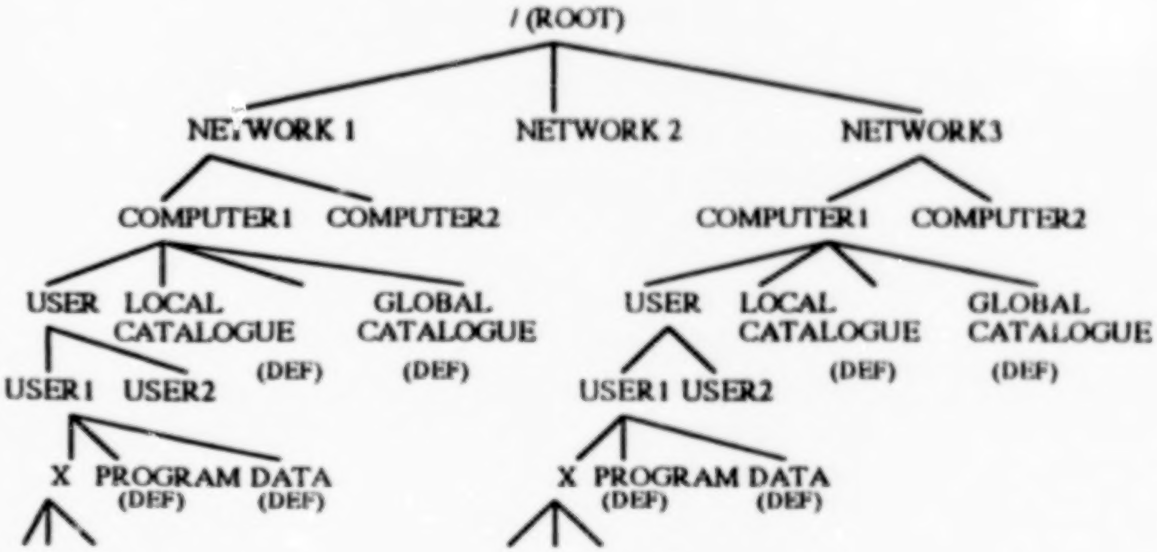
SPACE & EARTH SCIENCES DIRECTORATE

CODE 600

DISTRIBUTED ACCESS VIEW INTEGRATED DATABASE (DAVID) SYSTEM

GSFC

USER'S VIEW OF DAVID



/ (ROOT)

NETWORK 1

NETWORK 2

NETWORK3

COMPUTER1

COMPUTER2

COMPUTER1

COMPUTER2

USER

LOCAL CATALOGUE (DEF)

GLOBAL CATALOGUE (DEF)

USER

LOCAL CATALOGUE (DEF)

GLOBAL CATALOGUE (DEF)

USER1

USER2

USER1

USER2

X

PROGRAM (DEF)

DATA (DEF)

X

PROGRAM (DEF)

DATA (DEF)

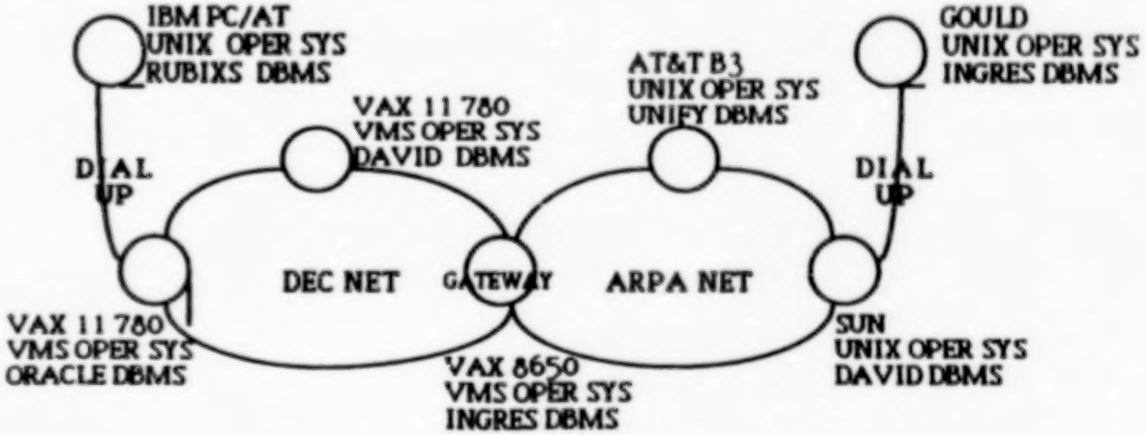
DRO8516L A-2

SPACE & EARTH SCIENCES DIRECTORATE

CODE 600

DISTRIBUTED ACCESS VIEW INTEGRATED DATABASE (DAVID) SYSTEM

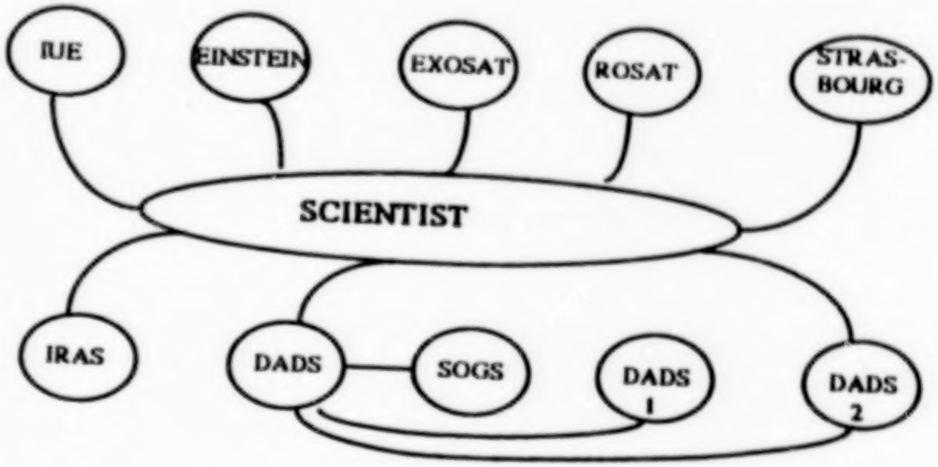
PHASE I: DEMONSTRATION SUBSYSTEM


IBM PC/AT
UNIX OPER SYS
RUBIXS DBMS
DIAL UP
VAX 11 780
VMS OPER SYS
DAVID DBMS
AT&T B3
UNIX OPER SYS
UNIFY DBMS
GOULD
UNIX OPER SYS
INGRES DBMS
DIAL UP
DEC NET
GATEWAY
ARPA NET
VAX 11 780
VMS OPER SYS
ORACLE DBMS
VAX 8650
VMS OPER SYS
INGRES DBMS
SUN
UNIX OPER SYS
DAVID DBMS

DISTRIBUTED ACCESS VIEW INTEGRATED DATABASE (DAVID) SYSTEM

GSFC

PHASE II. THE ASTROPHYSICS DATA SYSTEM


IUE
EINSTEIN
EXOSAT
ROSAT
STRAS-BOURG
SCIENTIST
IRAS
DADS
SOGS
DADS 1
DADS 2

SPACE & EARTH SCIENCES DIRECTORATE CODE 600	DISTRIBUTED ACCESS VIEW INTEGRATED DATABASE (DAVID) SYSTEM	GSFC

DAVID™ DBMS

SYSTEM OVERVIEW

TERMINAL INTERFACE PACKAGE 1	GSQL TRANSLATION PACKAGE 2	COMMUNICATION PACKAGE 3	DAVID GSQL PRIMITIVE PACKAGE 4	HOST LANGUAGE INTERFACE PACKAGE 5
GENERIC CLUSTER ACCESS PACKAGE 6	FILE ACCESS PACKAGE 7	DISTRIBUTED DICTIONARY PACKAGE 8	RESILIENT GSQL PRIMITIVE PACKAGE 9	CONSTRAINT REALIZATION PACKAGE A
VIEW INTEGRATION PACKAGE B	CONCURRENCY CONTROL PACKAGE C	SCHEDULER D	INSTALLATION & BACKUP PACKAGE E	UTILITIES PACKAGE F

DRO0536 A 2

INTELLIGENT DATA MANAGEMENT PROCESSES

NOVEMBER 19, 1986

W. J. CAMPBELL

NASA

HYPOTHESIS FOR RESEARCH

NATIONAL
SPACE SCIENCE
DATA CENTER

IT IS POSSIBLE TO DEVELOP ADVANCED DATA MANAGEMENT SYSTEMS AND SERVICES THAT WILL FUNCTION LIKE A COLLECTIVE GROUP OF KNOWLEDGEABLE DATABASE EXPERTS AND SUPPORT ENGINEERS SUCH THAT THE SYSTEM CAN SUPPORT A WIDE RANGE OF USERS THAT HAVE MINIMAL DATABASE UNDERSTANDING AND EXPERIENCE

NASA

BASIS FOR THE HYPOTHESIS

NATIONAL SPACE SCIENCE DATA CENTER

- A DATABASE HAS A FINITE AND RATHER LIMITED KNOWLEDGE SPACE WHICH MAKES THE APPLICATION OF EXPERT SYSTEMS AND RELATED AI TECHNOLOGIES POSSIBLE

- MANY OPERATIONS THAT ARE PERFORMED BY EXPERT DATABASE USERS ARE BASED ON HEURISTIC OR PROCEDURAL KNOWLEDGE

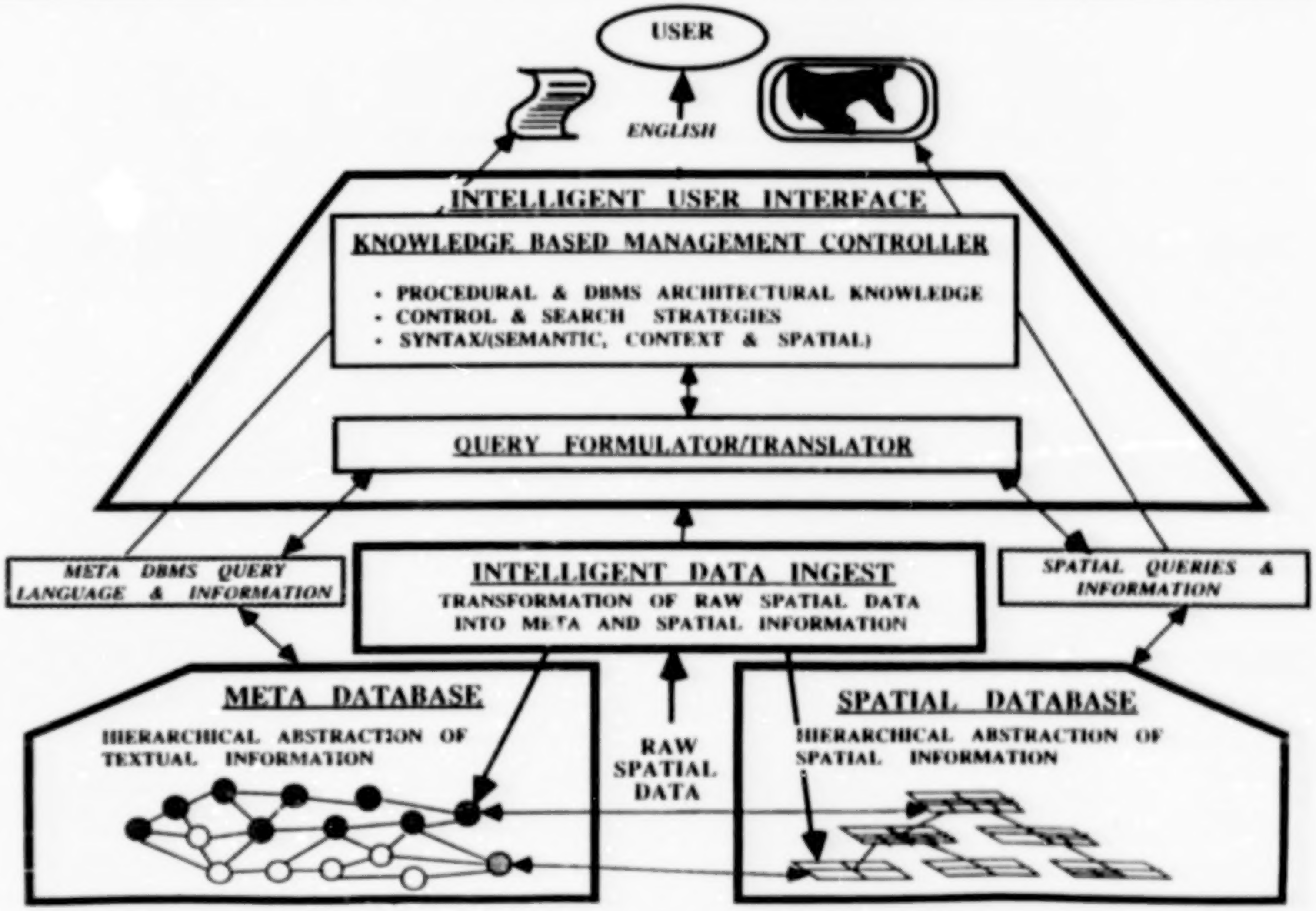
- ALTHOUGH MOST OPERATIONAL DATABASES CAN HAVE A LARGE NUMBER OF DATA OBJECTS, THE ACTUAL NUMBER OF USEFUL OBJECTS IS USUALLY RATHER LIMITED, AND FOR ANY SPECIFIC APPLICATION DOMAIN IS SMALL

- EXPERT SYSTEMS WILL ALLOW THE IMPLEMENTATION OF MULTIPLE DATABASE VIEWS AND SUPPORTING SEARCH STRATEGIES FOR A BROAD RANGE OF DOMAIN SPECIFIC APPLICATIONS

NASA

IDM CONCEPT

NATIONAL SPACE SCIENCE DATA CENTER



USER

ENGLISH

INTELLIGENT USER INTERFACE

KNOWLEDGE BASED MANAGEMENT CONTROLLER

- PROCEDURAL & DBMS ARCHITECTURAL KNOWLEDGE
- CONTROL & SEARCH STRATEGIES
- SYNTAX/(SEMANTIC, CONTEXT & SPATIAL)

QUERY FORMULATOR/TRANSLATOR

META DBMS QUERY LANGUAGE & INFORMATION

INTELLIGENT DATA INGEST

TRANSFORMATION OF RAW SPATIAL DATA INTO META AND SPATIAL INFORMATION

SPATIAL QUERIES & INFORMATION

META DATABASE

HIERARCHICAL ABSTRACTION OF TEXTUAL INFORMATION

RAW SPATIAL DATA

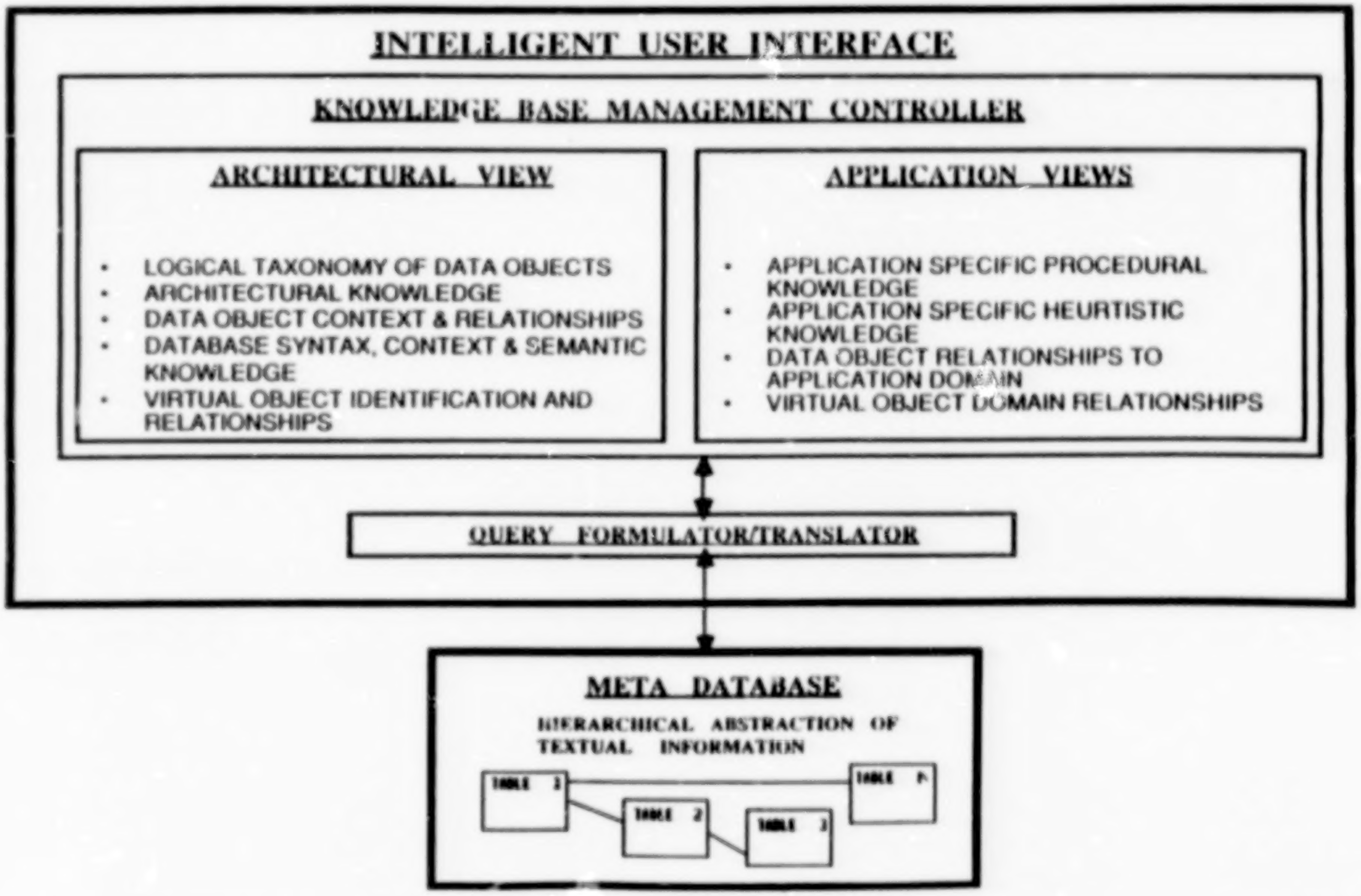
SPATIAL DATABASE

HIERARCHICAL ABSTRACTION OF SPATIAL INFORMATION

NASA

OVERVIEW OF INTELLIGENT USER INTERFACE DESIGN

NATIONAL
SPACE SCIENCE
DATA CENTER



INTELLIGENT USER INTERFACE

KNOWLEDGE BASE MANAGEMENT CONTROLLER

ARCHITECTURAL VIEW

- LOGICAL TAXONOMY OF DATA OBJECTS
- ARCHITECTURAL KNOWLEDGE
- DATA OBJECT CONTEXT & RELATIONSHIPS
- DATABASE SYNTAX, CONTEXT & SEMANTIC KNOWLEDGE
- VIRTUAL OBJECT IDENTIFICATION AND RELATIONSHIPS

APPLICATION VIEWS

- APPLICATION SPECIFIC PROCEDURAL KNOWLEDGE
- APPLICATION SPECIFIC HEURTISTIC KNOWLEDGE
- DATA OBJECT RELATIONSHIPS TO APPLICATION DOMAIN
- VIRTUAL OBJECT DOMAIN RELATIONSHIPS

QUERY FORMULATOR/TRANSLATOR

META DATABASE

HIERARCHICAL ABSTRACTION OF TEXTUAL INFORMATION

NASA

EXPECTED IUI CAPABILITIES

NATIONAL
SPACE SCIENCE
DATA CENTER

- IT CAN FACILITATE UNDERSTANDING BY SPECIFYING THE CONTENTS AND MEANINGS OF A COLLECTION OF DATA AS WELL AS THE RELATION BETWEEN OBJECTS WITHIN THE DATA
- IT CAN SUPPORT APPROXIMATE REASONING BY INFERING CONCLUSIONS THAT ARE NOT EXPLICITELY STATED BY THE USER, SUCH AS AN IMPRECISE QUERY
- FUZZY CONCEPTS CAN BE EXPRESSED BY USING A FUZZY QUERY
- IT CAN HANDLE INFORMATION DEMAND FOR WHICH THE DATABASE IS USED ROUTINELY, RAPIDLY AND EFFICIENTLY WITHOUT ANY UNDERSTANDING BY THE USER
- CAN COMMUNICATE WITH THE USER IN PLAIN ENGLISH TEXT

NASA

EXPECTED IUI CAPABILITIES

(CONTINUED)

NATIONAL
SPACE SCIENCE
DATA CENTER

- CAN SERVE AS THE SEMANTIC BASIS FOR UNDERSTANDING THE USER'S DATA NEEDS IN CONJUNCTION WITH THE NATURAL LANGUAGE QUERY PROCESSOR
- CAN PROVIDE A LOGICAL REPRESENTATION OF THE DATABASE ARCHITECTURE AND INFORMATION STORED IN THE DATABASE TO THE CASUAL USER
- CAN FACILITATE AN UNDERSTANDING AND IDENTIFICATION OF DATABASE INFORMATION FROM THE DATABASE OPERATIONAL VIEW TO THE USER RELATED VIEWS BY USING CONCEPTUAL GRAPHS

NASA

IUIS DESIGN CONCEPTS

**NATIONAL
SPACE SCIENCE
DATA CENTER**

- REPRESENT THE META DATABASE BY VIEWS THAT ARE FUNCTIONALLY APPROPRIATE TO THE USER AND AT THE SAME TIME OPERATIONALLY NECESSARY TO REPRESENT THE ACTUAL DATABASE

- CREATE TWO GENERAL TYPES OF VIEWS OF THE META DATABASE:
  - AN ARCHITECTURAL VIEW
  - MULTIPLE APPLICATION VIEWS

- DEVELOP PROCESSES BASED ON CONCEPTUAL GRAPHS THAT WILL ALLOW LOGICAL TRANSLATION BETWEEN THE VARIOUS DATABASE VIEWS

NASA

IUI DESIGN APPROACH

NATIONAL SPACE SCIENCE DATA CENTER

- USE EXPERT SYSTEMS TO SUPPORT THE DEVELOPMENT OF THE INTELLIGENT USER INTERFACE
- REPRESENT THE APPLICATION VIEWS WITH GOAL DIRECTED BACKWARD CHAINING PROCESSES THAT CAPTURES EXPERT HEURISTIC AND PROCEDURAL KNOWLEDGE
- REPRESENT THE ARCHITECTURAL VIEW USING A FRAME BASED CONCEPT THAT WILL SUPPORT OBJECT ORIENTED RELATIONSHIPS AND INHERITANCE
- USE CONCEPTUAL GRAPHS TO TRANSLATE BETWEEN THE DATABASE'S MEANING AND THE USER'S MEANING OF DATA OBJECTS AND SETS OF DATA OBJECTS
- PROVIDE A LOGICAL LINK BETWEEN INFORMATION CONTAINED IN THE VIEWS OF THE META DATABASE AS WELL AS BETWEEN THE META AND THE SPATIAL DATABASE USING CONCEPTUAL GRAPHS

C-4

NASA

REPRESENTATION OF A CONCEPTUAL GRAPH

NATIONAL
SPACE SCIENCE
DATA CENTER

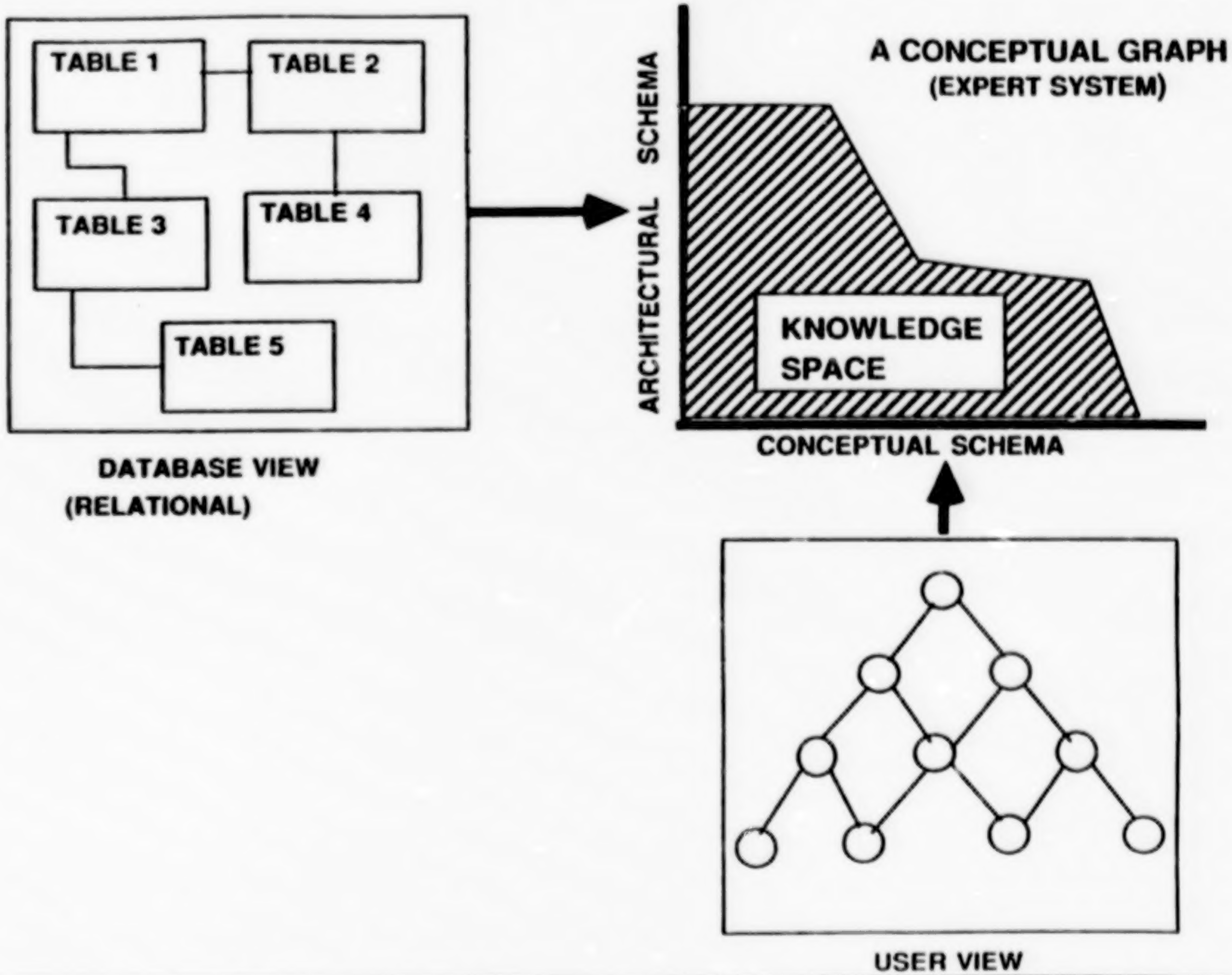


TABLE 1
TABLE 2
TABLE 3
TABLE 4
TABLE 5

DATABASE VIEW
(RELATIONAL)

A CONCEPTUAL GRAPH
(EXPERT SYSTEM)

ARCHITECTURAL SCHEMA

KNOWLEDGE
SPACE

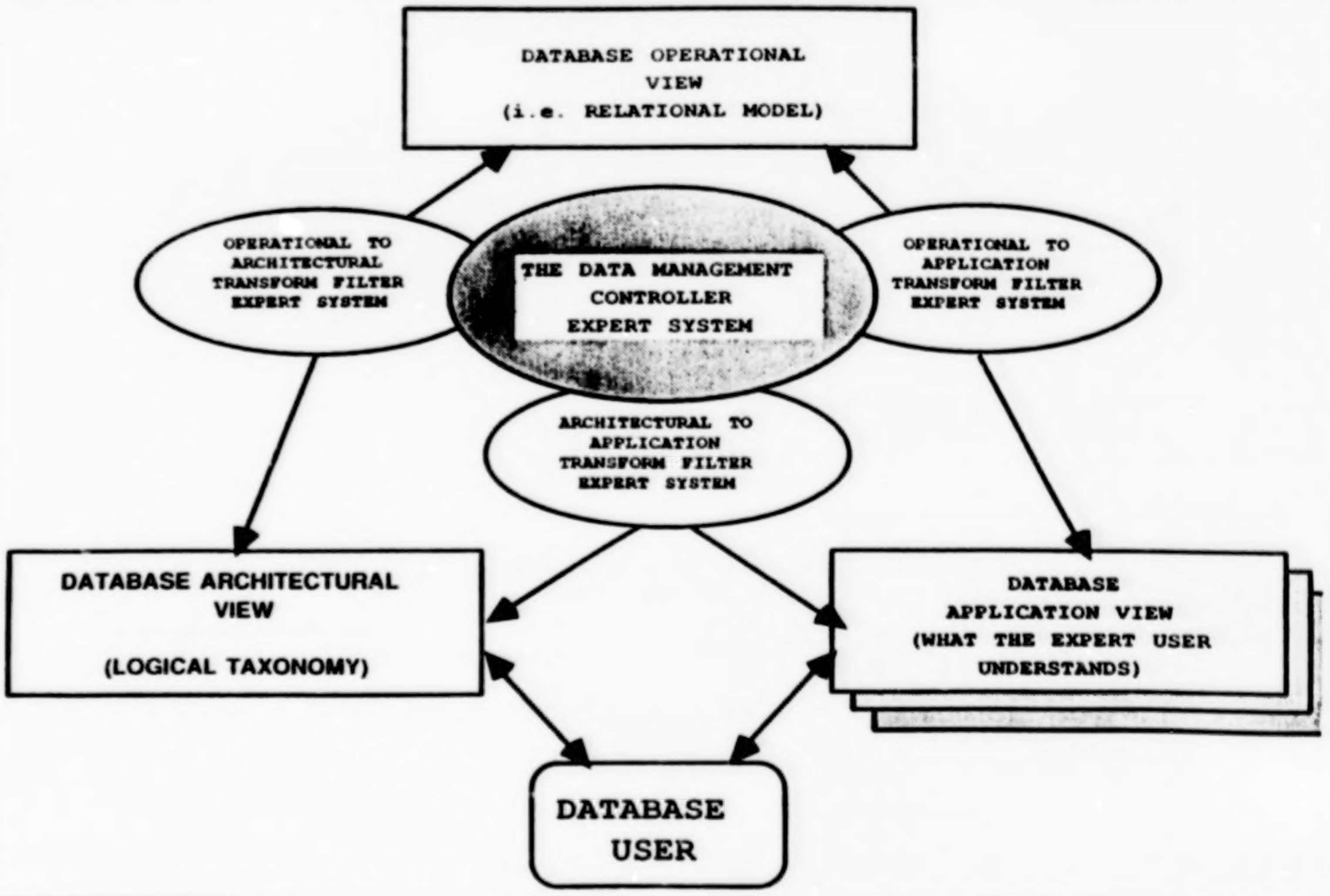
CONCEPTUAL SCHEMA

USER VIEW

NASA

MODEL OF THE IUI CONCEPT

NATIONAL SPACE SCIENCE DATA CENTER



DATABASE OPERATIONAL VIEW (i.e. RELATIONAL MODEL)

OPERATIONAL TO ARCHITECTURAL TRANSFORM FILTER EXPERT SYSTEM

THE DATA MANAGEMENT CONTROLLER EXPERT SYSTEM

OPERATIONAL TO APPLICATION TRANSFORM FILTER EXPERT SYSTEM

ARCHITECTURAL TO APPLICATION TRANSFORM FILTER EXPERT SYSTEM

DATABASE ARCHITECTURAL VIEW

(LOGICAL TAXONOMY)

DATABASE APPLICATION VIEW (WHAT THE EXPERT USER UNDERSTANDS)

DATABASE USER

VIRTUAL OBJECTS

THERE IS A CLASS OF DATABASE OBJECTS THAT THE EXPERT USER KNOWS ABOUT BUT THAT DOES NOT EXIST IN THE DATABASE CALLED VIRTUAL OBJECTS (SOMETIMES CALLED INFERED OBJECTS)

VIRTUaL OBJECTS ARE THE RESULT OF THE CLUSTERING OF SEVERAL TOUPLES (REAL DATABASE OBJECTS) TO FORM A NEW OBJECT

PRESENTLY, VIRTUAL OBJECTS CAN ONLY BE IDENTIFIED AND MANAGED WITH A KNOWLEDGE BASE

VIRTUAL OBJECTS ARE BEST REPRESENTED IN THE ARCHITECTURAL VIEW OF THE DATABASE, HOWEVER, THEY ARE MOST CERTAINLY USED IN THE APPLICATION VIEWS

THE IMPORTANCE OF VIRTUAL OBJECTS CAN NOT BE OVER EMPHASISED BECAUSE THEY REPRESENT INFORMATION THAT THE DATA CONTAINS THAT DOES NOT EXIST EXPLICIATELY IN THE DATABASE BUT IS THE RESULT OF AN EXPERTS KNOWLDEGE ABOUT THE DATABASE

NASA

INTELLIGENT DATA MANAGEMENT
FOCUS FY 86

NATIONAL
SPACE SCIENCE
DATA CENTER

- FORMULATE, DESIGN AND PROTOTYPE PHASE ONE INTELLIGENT USER INTERFACE WITH THE FOLLOWING COMPONENTS:
    - KNOWLEDGE BASED MANAGEMENT CONTROLLER
    - QUERY FORMULATOR/TRANSLATOR
- PROTOTYPE INITIAL KNOWLEDGE BASED MANAGEMENT CONTROLLER USING EXPERT SYSTEM TECHNOLOGIES
- DESIGN, PROTOTYPE AND INTERFACED A QUERY FORMULATOR/ TRANSLATOR TO THE KNOWLEDGE BASED MANAGEMENT CONTROLLER
- DEMONSTRATE THE PROTOTYPE SYSTEM WITH OPERATIONAL SCIENTIFIC DATABASE (CRUSTAL DYNAMICS PROJECT)

CAPABILITIES OF INITIAL KNOWLEDGE BASED MGMT CONTROLLER

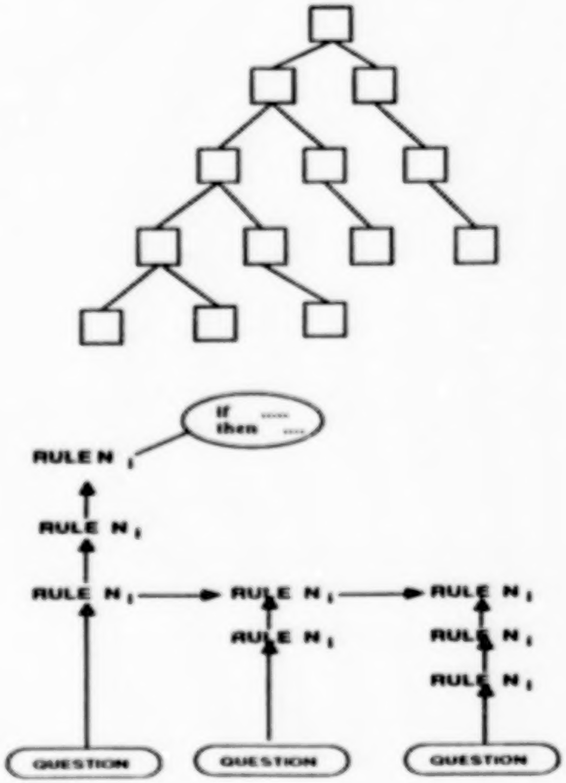
- TWO DATABASE VIEWS REPRESENTED: ARCHITECTURAL & APPLICATION
- ARCHITECTURAL VIEW BASED ON A MODIFIED HIERARCHY REPRESENTATION WITH LIMITED CONTEXTUAL INFORMATION
- SINGLE APPLICATION VIEW
- BOTH KNOWLEDGE BASE AND DOMAIN FACTS CONTAINED IN RULES AND SUPPORTING TABLES
- REASONING LIMITED TO GOAL DIRECTED BACKWARD CHAINING
- SINGLE LINE OF REASONING
- PROTOTYPE LIMITED BY EXPERT SYSTEM AND QUERY PROCESSING TOOLS AND COMPUTER RESOURCES

NASA

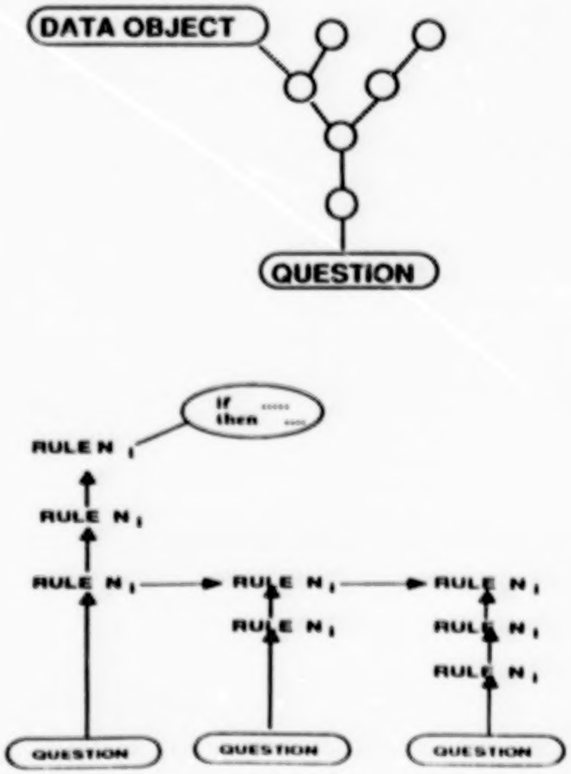
INITIAL KBM CONTROLLER DESIGN

NATIONAL
SPACE SCIENCE
DATA CENTER

DATABASE ARCHITECTURE REPRESENTED IN A LOGICAL HIERARCHY SUPPORTED BYMINIMUM RELATIONSHIP BETWEEN DATA OBJECTS


if ..... then .....
RULE N i
RULE N i
RULE N i
RULE N i
RULE N i
RULE N i
RULE N i
RULE N i
QUESTION
QUESTION
QUESTION

APPLICATION VIEW IS RULE BASED GOAL DRIVEN


DATA OBJECT
QUESTION
if ..... then .....
RULE N i
RULE N i
RULE N i
RULE N i
RULE N i
RULE N i
RULE N i
RULE N i
QUESTION
QUESTION
QUESTION

NASA

RESEARCH RESULTS

NATIONAL
SPACE SCIENCE
DATA CENTER

- IT IS NECESSARY TO REPRESENT THE DATABASE WITH MULTIPLE VIEWS INCLUDING BOTH A LOGICAL AND OPERATIONAL VIEWS
- IT IS POSSIBLE TO CREATE THE DATABASE VIEWS USING AN EXPERT SYSTEM
- EXPERT SYSTEMS PROVIDE A MEANS OF REPRESENTING BOTH THE OPERATIONAL CONTEXT AND SEMANTICS TO A DATABASE AS WELL AS ITS SYNTAX
- IT IS BETTER TO REASON IN THE EXPERT SYSTEM IN THE LANGUAGE DOMAIN OF THE USER
- THE USE OF CONCEPTUAL GRAPHS ALLOW THE TRANSLATION BETWEEN THE VARIOUS DATABASE VIEWS
- ENGLISH IS A VERY SYNTACTICALLY COMPACT LANGUAGE FOR REASONING ABOUT DATABASES
- THE SYSTEM MUST INCLUDE VIRTUAL OBJECTS WHICH ARE THE RESULT OF THE CLUSTERING OF REAL DATABASE OBJECTS

NASA

INTELLIGENT DATA MANAGEMENT RESEARCH FOCUS FOR FY87

NATIONAL SPACE SCIENCE DATA CENTER

- THE DEVELOPMENT OF AN ADVANCED KNOWLEDGE BASED MANAGEMENT CONTROLLER THAT SUPPORTS:
    - DATABASE ARCHITECTURAL VIEW BASED ON A LOGICAL TAXONOMY USING A FRAME BASED REPRESENTATION SCHEMA
    - THE IDENTIFICATION OF DATA OBJECT CONTEXT IN ARCHITECTURAL VIEW UTILIZING PARENT/DAUGHTER RELATIONSHIP INHERENT IN THE FRAME BASED DATA STRUCTURE
- INCLUSION OF GRAPHICAL DATA OBJECTS IN FRAME SCHEMA TO SUPPORT MANAGEMENT AND UNDERSTANDING OF HIGH LEVEL DATA OJBECTS (E.G., MAPS, DIAGRAMS)
- DEVELOPMENT OF A GENERALIZED QUESTION GENERATOR USING CONTEXT AND GOAL INFERED INFORMATION
- PROTOTYPE A SECOND GENERATION INTELLIGENT USER INTERFACE FOR AN OPERATIONAL SCIENTIFIC DATABASE

NASA

CAPABILITIES OF ADVANCED KNOWLEDGE BASED MGMT CONTROLLER

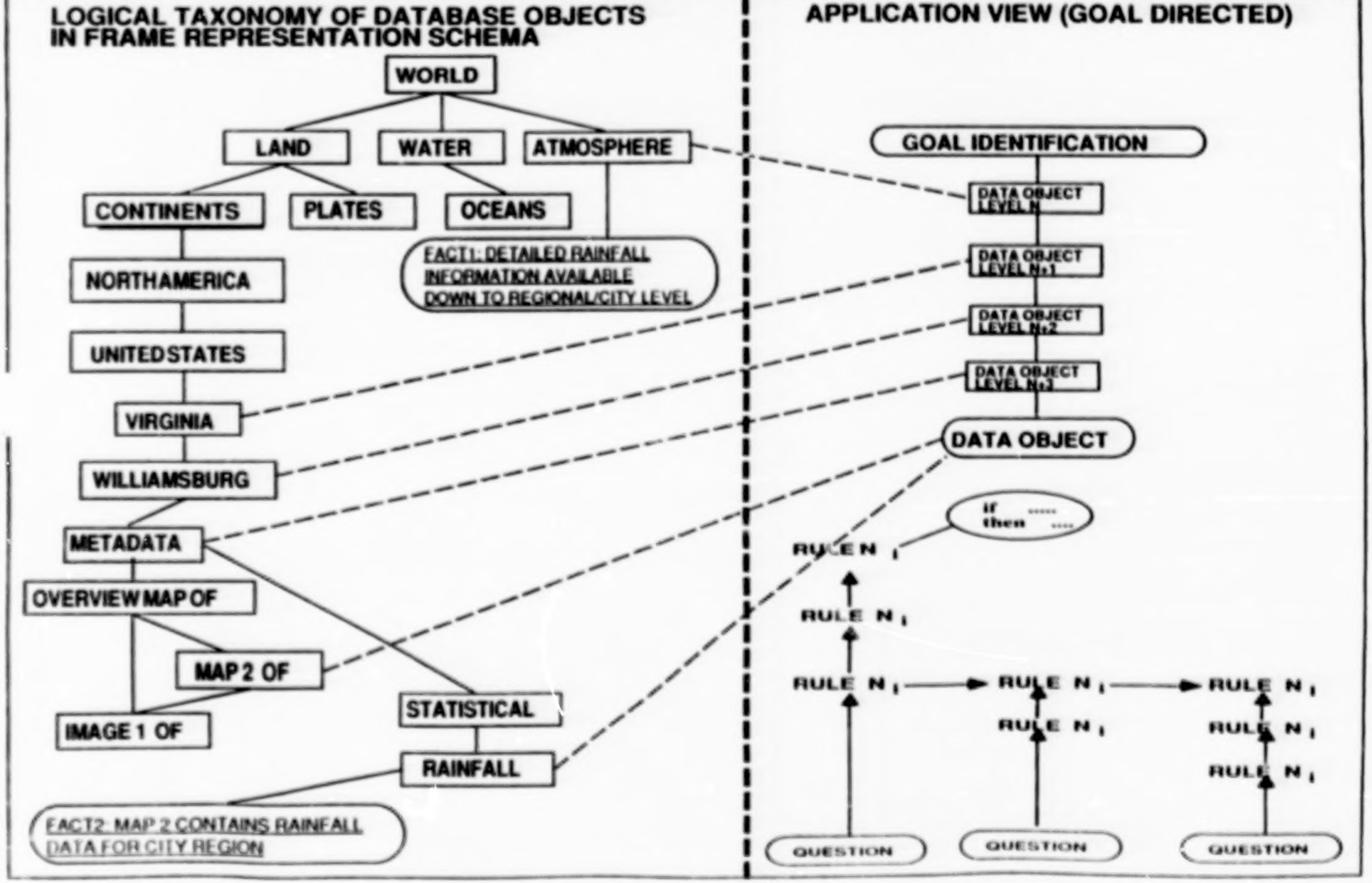
NATIONAL
SPACE SCIENCE
DATA CENTER

- MULTIPLE ARCHITECTURAL AND APPLICATION VIEWS REPRESENTED
- ARCHITECTURAL VIEW IS A HIERARCHICAL REPRESENTATION OF THE DATABASE OBJECTS WITH FULL CONTEXT RELATIONSHIP SUPPORTED BY FRAME BASED DATA STRUCTURE WITH INHERITANCE
- APPLICATION VIEW DOMAIN IS A SUBSET OF FULL ARCHITECTURAL VIEW DOMAIN WITH APPLICATION SPECIFIC RULE BASE
- DATA OBJECT SEARCH SUPPORTED BY CONCURRENT FORWARD AND BACKWARD REASONING
- PSUEDO PARALLEL DATA OBJECT IDENTIFICATION AND SELECTION (CONCURRENT MULIPLE LINES OF REASONING) SUPPORTED BY MULTIWORLD OPERATIONS
- DATA STRUCTURE SUPPORTS BOTH META AND SPATIAL DATA OBJECTS
- NEAR REAL-TIME CONSTRUCTION OF OPTIMAL SEARCH NETWORK USING PATTERN RECOGNITION PROCESSES THAT SUPPORT A DYNAMIC DATABASE ARCHITECTURE FOR APPLICATION SPECIFIC PROBLEMS

NASA

DESIGN OVERVIEW OF ADVANCED KNOWLEDGE BASE MGMT CONTROLLER

**NATIONAL
SPACE SCIENCE
DATA CENTER**


LOGICAL TAXONOMY OF DATABASE OBJECTS IN FRAME REPRESENTATION SCHEMA
WORLD
LAND
WATER
ATMOSPHERE
CONTINENTS
PLATES
OCEANS
FACT1: DETAILED RAINFALL INFORMATION AVAILABLE DOWN TO REGIONAL/CITY LEVEL
NORTHAMERICA
UNITEDSTATES
VIRGINIA
WILLIAMSBURG
METADATA
OVERVIEW MAP OF
MAP 2 OF
IMAGE 1 OF
STATISTICAL
RAINFALL
FACT2: MAP 2 CONTAINS RAINFALL DATA FOR CITY REGION
APPLICATION VIEW (GOAL DIRECTED)
GOAL IDENTIFICATION
DATA OBJECT LEVEL N
DATA OBJECT LEVEL N+1
DATA OBJECT LEVEL N+2
DATA OBJECT LEVEL N+3
DATA OBJECT
if ..... then .....
RULE N_i
RULE N_i
RULE N_i
RULE N_i
RULE N_i
RULE N_i
RULE N_i
RULE N_i
QUESTION
QUESTION
QUESTION

PERFORMANCE COMPARISONS

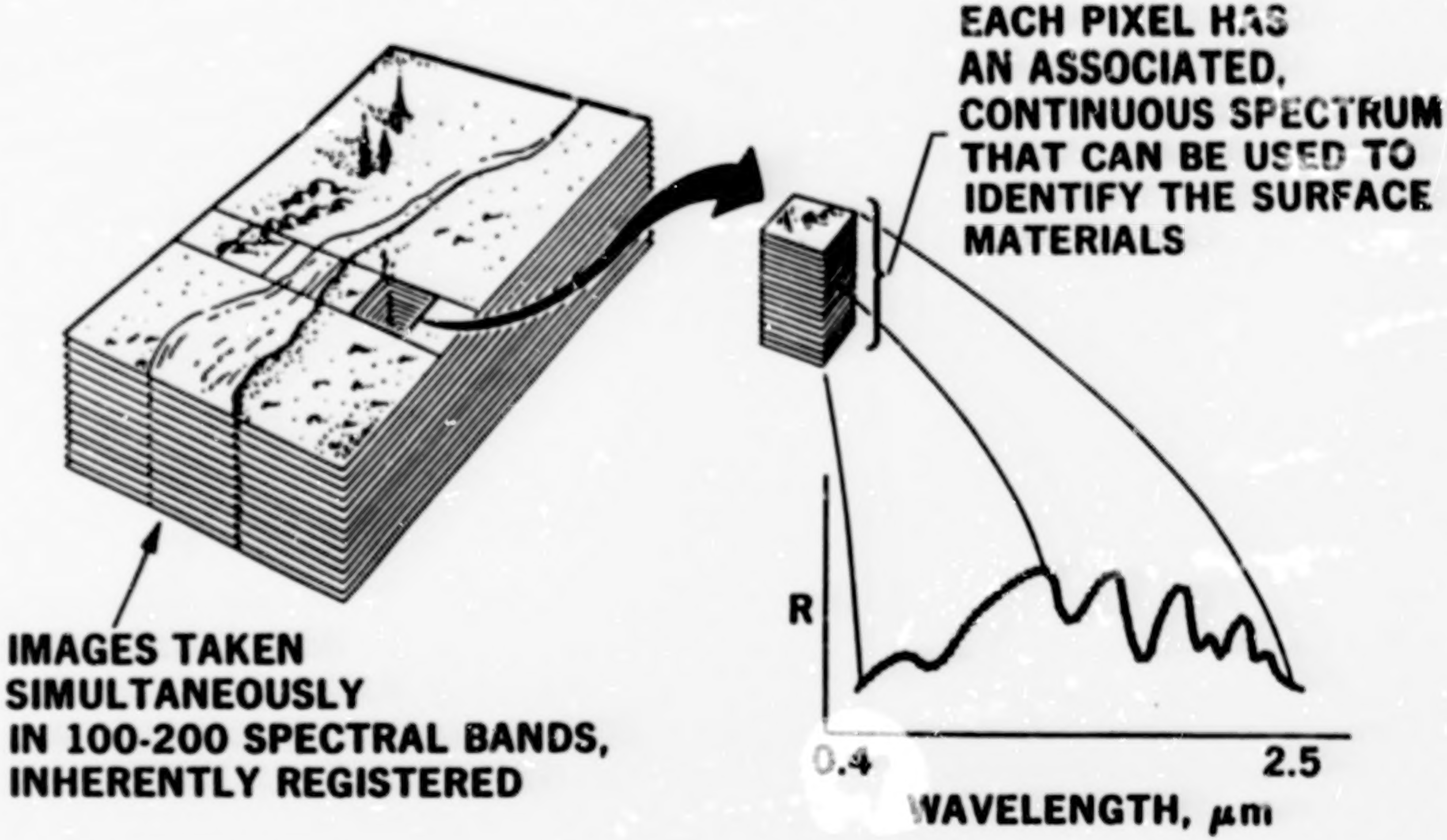
INITIAL	ADVANCED
DATABASE ARCHITECTURE	
• DATABASE ARCHITECTURE AND FACTS INCLUDED IN RULE BASE	• ARCHITECTURE REPRESENTED IN A LOGICAL TAXONOMY OF FRAMES/OBJECTS • APPLICATION VIEWS AUTOMATICALLY CREATED BY SELECTING SPECIFIC GOALS AND OBJECTS • CONTEXT SUPPORTED BY INHERITANCE IN DATA STRUCTURE • PARALLEL DATA SEARCH SUPPORTED BY MULTIWORLD REASONING
REASONING PROCESS	
• GOAL DRIVEN BACKWARD CHAINING	• GOAL DRIVEN BACKWARD CHAINING AND DATA DRIVEN FORWARD CHAINING CONCURRENTLY
QUERY FORMULATION	
• QUERY FORMULATION TO DBMS SUPPORTED BY NATURAL LANGUAGE QUERY PROCESSOR BETWEEN EXPERT SYSTEM AND DBMS	• QUERY FORMULATION TO DBMS SUPPORTED BY NATURAL LANGUAGE QUERY PROCESSOR BETWEEN EXPERT SYSTEM AND DBMS

AN EXPERT SYSTEM FOR THE ANALYSIS OF IMAGING SPECTROMETER DATA

JPL

GARY C . BORCHARDT

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

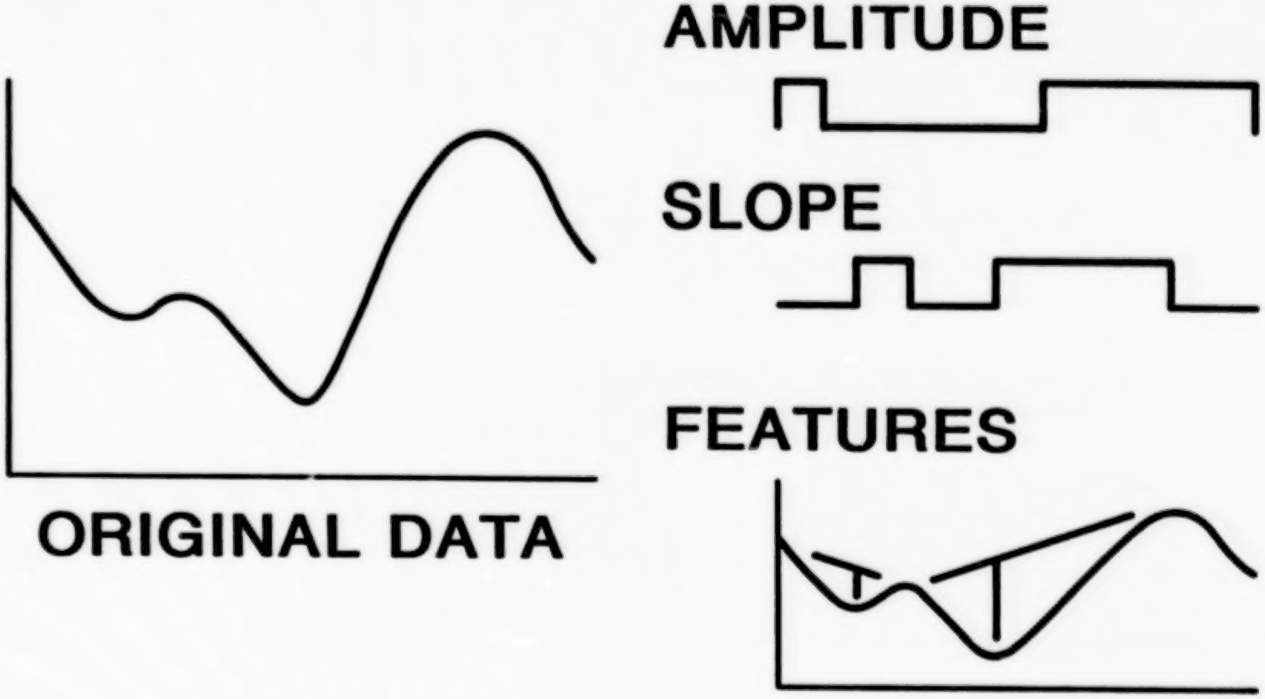


**EACH PIXEL HAS
AN ASSOCIATED,
CONTINUOUS SPECTRUM
THAT CAN BE USED TO
IDENTIFY THE SURFACE
MATERIALS**

**IMAGES TAKEN
SIMULTANEOUSLY
IN 100-200 SPECTRAL BANDS,
INHERENTLY REGISTERED**

JPL

REPRESENTATION OF SPECTRAL DATA


AMPLITUDE
SLOPE
FEATURES
ORIGINAL DATA

SUPPLEMENTARY INFORMATION

GEOLOGY/MINERALOGY:

MINERAL CLASSIFICATIONS, ABUNDANCE, STABILITY, INTERASSOCIATIONS.

FROM THE USER:

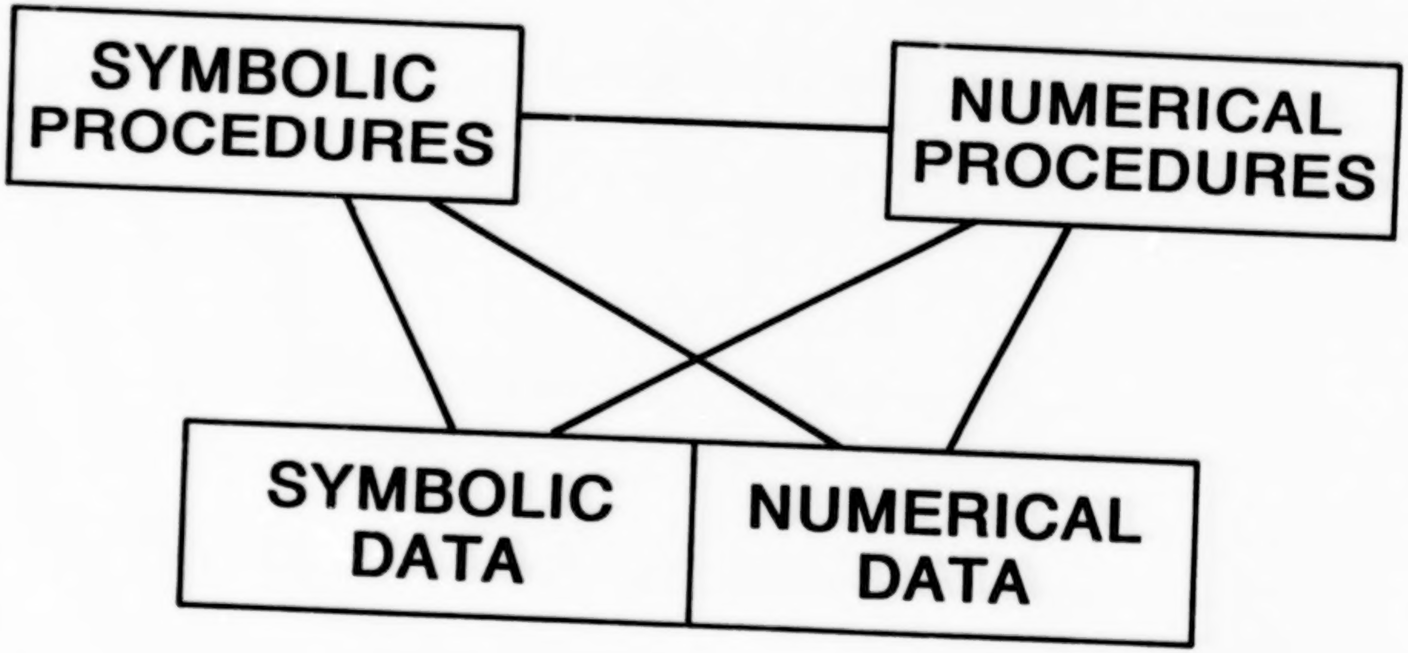
EXPECTATIONS, SUGGESTIONS.

FROM THE IMAGE DATA:

REGIONS OF SIMILARITY, ADJACENCIES.

JPL

STAR
(SIMPLE TOOL FOR AUTOMATED REASONING)



SYMBOLIC PROCEDURES

NUMERICAL PROCEDURES

SYMBOLIC DATA

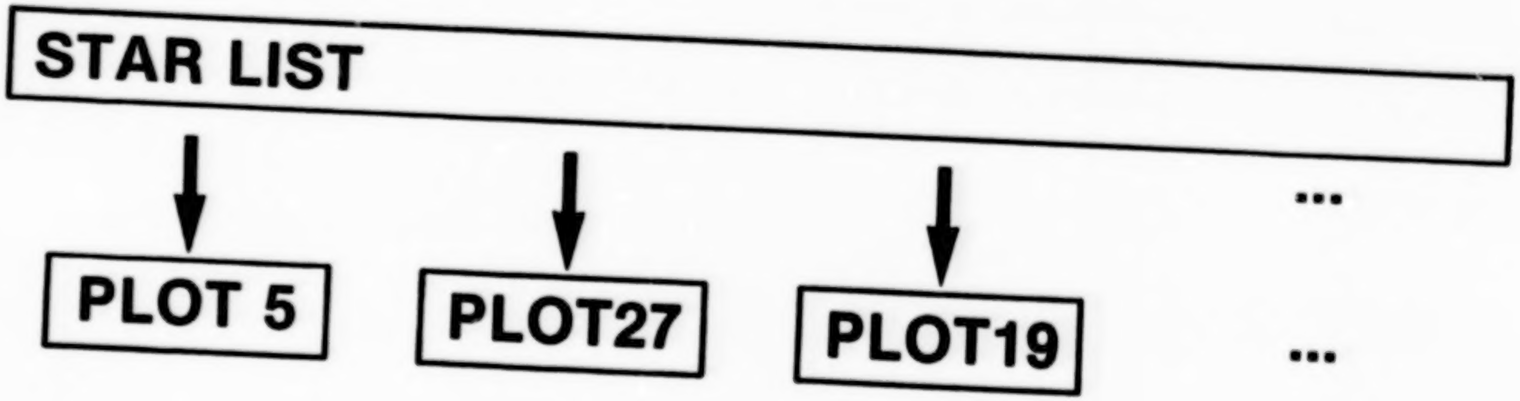
NUMERICAL DATA

STAR DATA STRUCTURES (UNITS)

NUMBERS	3, -27, 100.4,
TOKENS	ALUMINUM, MULTIPLY,
STRINGS	"measure of similarity",
LISTS	[0.7 0.6 0.5], [ ],
RECORDS	{time -> 8 value -> true },
EXPRESSIONS	+(2 3), append([2 3] [4 5]),
CONNECTIONS	^C_ROUTINE, ^HASH_TABLE.

CANDIDATES ([^PLOT5 ^PLOT27^PLOT19 ...]
^PLOT4 15 BINARY)

→ [^PLOT5 ^PLOT19 ...]


STAR LIST
...
PLOT 5
PLOT27
PLOT19
...

SAMPLE STAR CONSTRUCTS

```
{name -> BIOTITE
 member_of -> class
 subclass_of -> mica
 members -> [biotite_1 biotite_2]
 subclasses -> []
 abundance -> very_common
 stability -> unstable
 associated_with ->
      [muscovite orthoclase amphibole]
 }
```

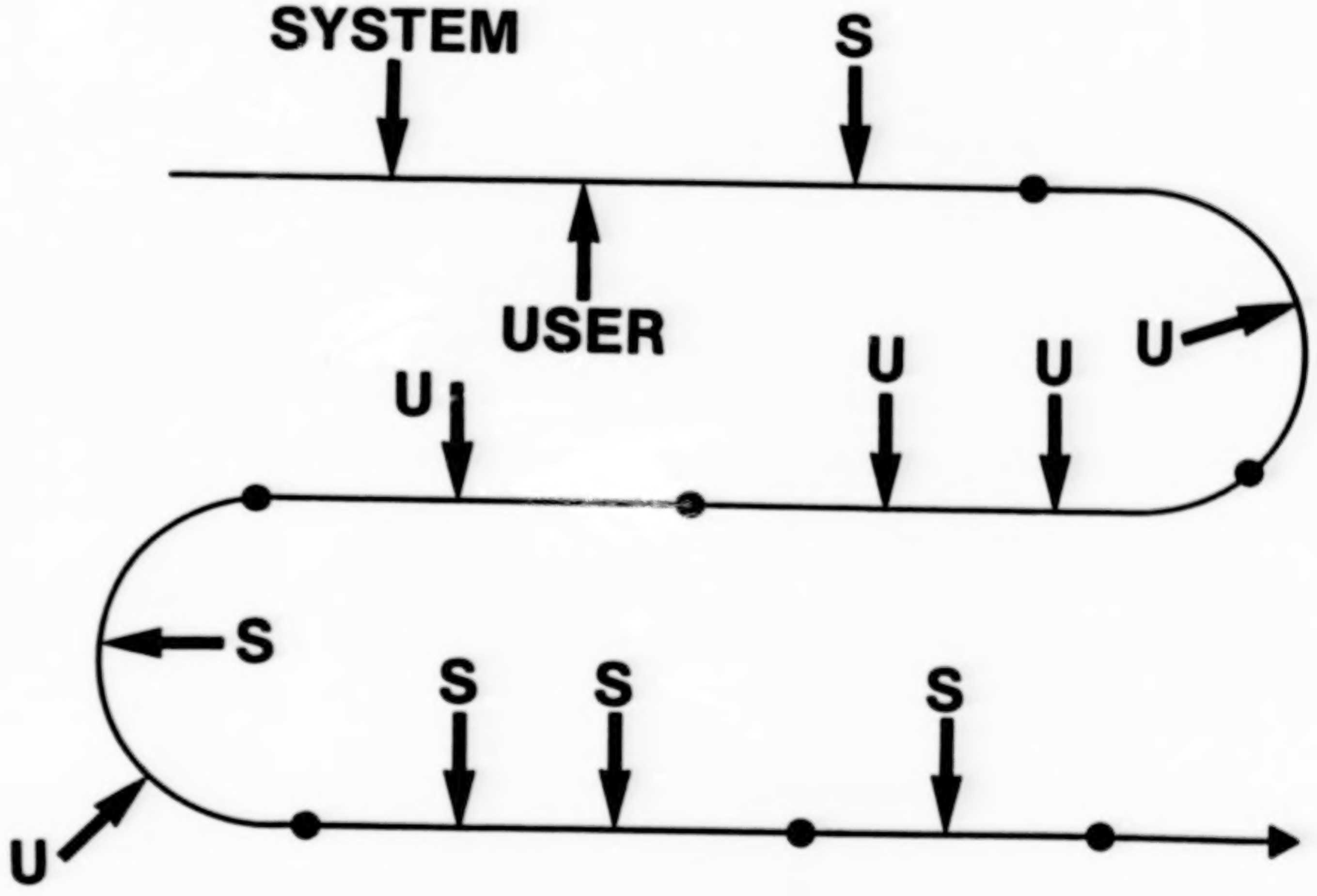
```
{name -> BIOTITE_1
 member_of -> biotite
 data -> ^PLOT_305460
 }
```

```
{name -> BIOTITE_2
 member_of -> biotite
 data -> ^PLOT_306520
 }
```

```
{name -> DISTANCE
 member_of -> c_application_function
 comment ->
     "      (^PLOT1 ^PLOT2 NUMBER1) => NUMBER
     "
     "      Measures the distance between the
     "encodings of two plots, ^PLOT1 and ^PLOT2.
     "The particular encoding method and distance
     "measure used is determined by the value of
     "NUMBER1."
 n_arguments -> 3
 algorithm -> ^C_DISTANCE_FUNCTION
 }
```

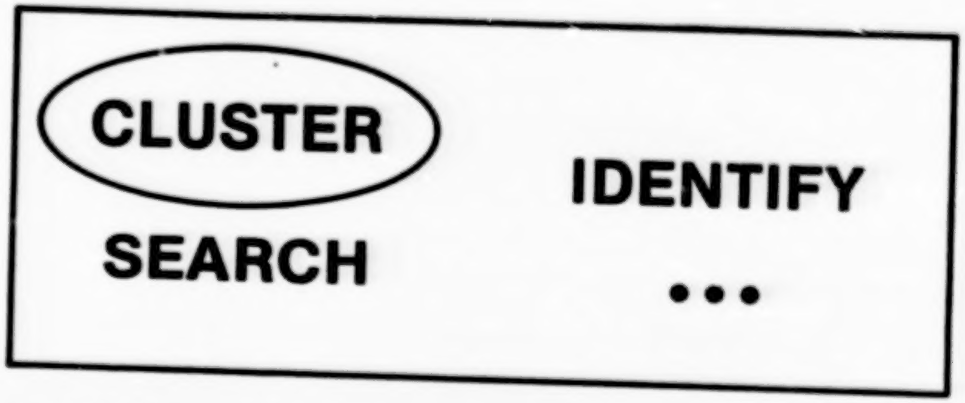
JPL

MIXED-INITIATIVE CONTROL


SYSTEM
S
USER
U
U
U
U
S
S
S
S
U

JPL

FOCUS OF ACTIVITY



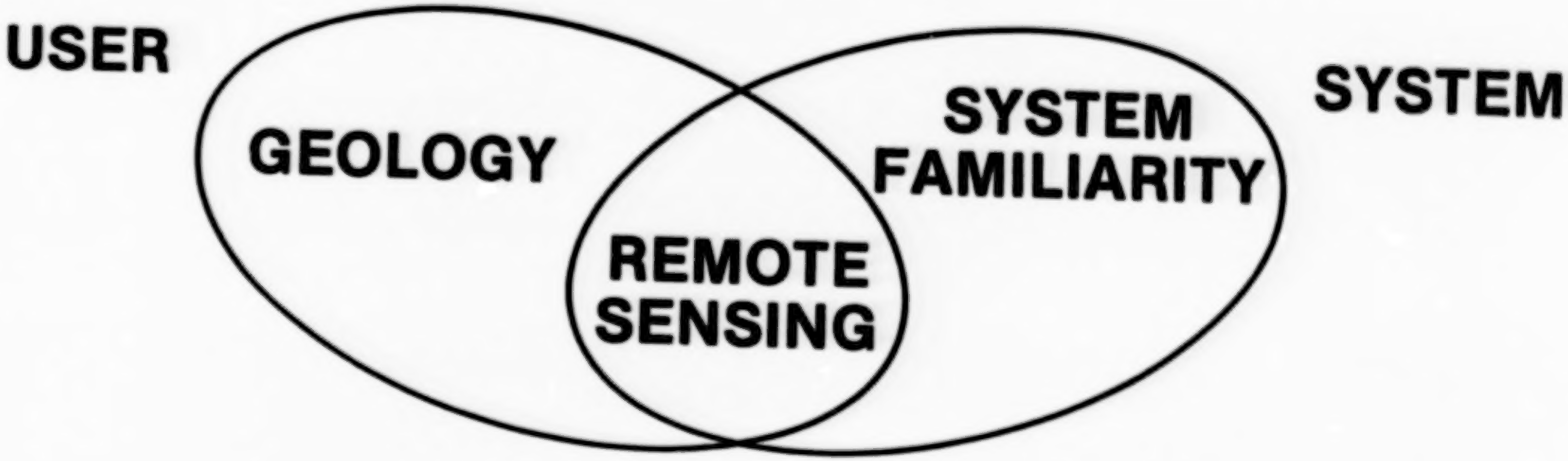
CLUSTER

SEARCH

IDENTIFY

• • •

DECISION AREAS OF INTEREST



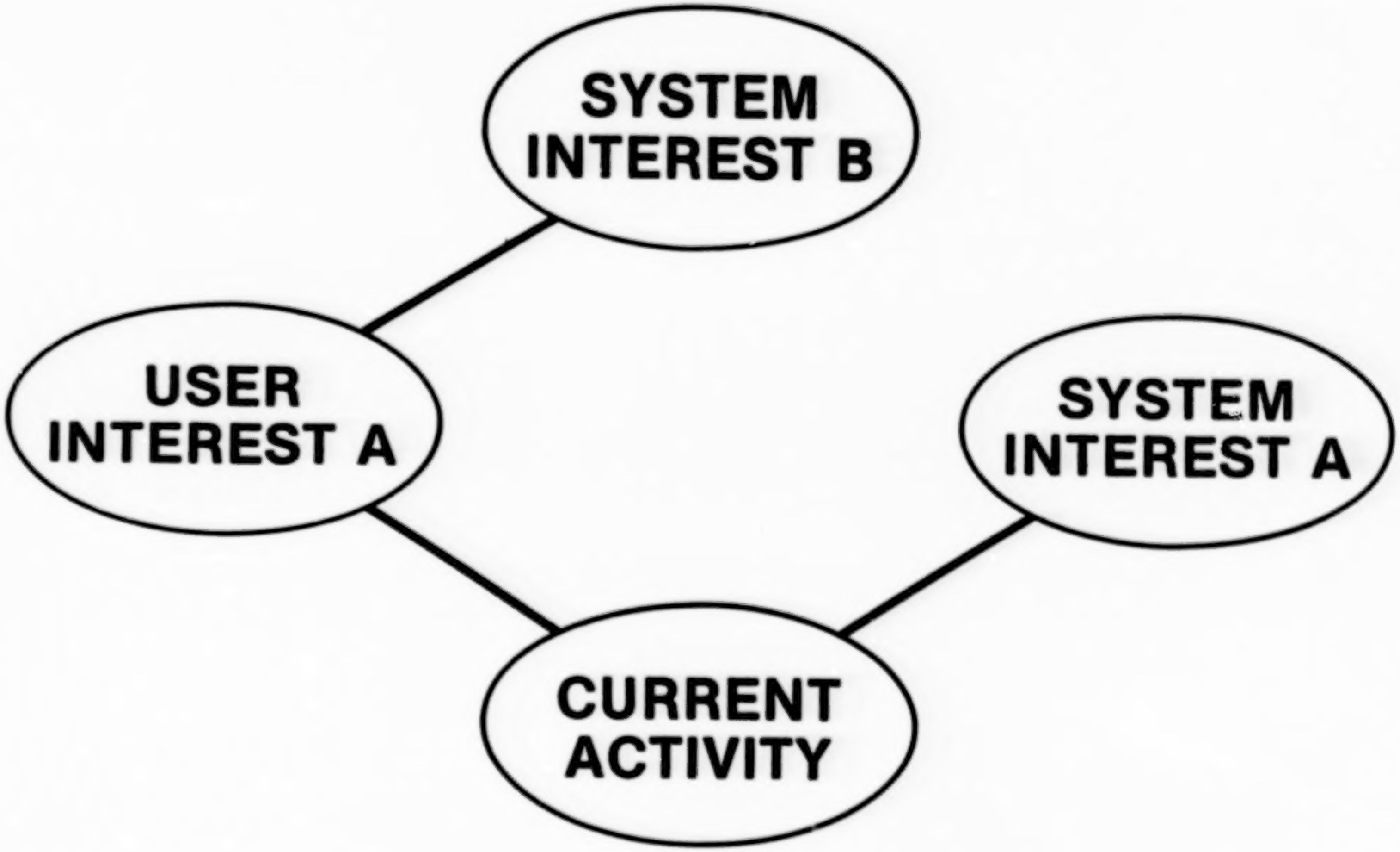
USER

GEOLOGY

REMOTE SENSING

SYSTEM FAMILIARITY

SYSTEM



SYSTEM INTEREST B

USER INTEREST A

SYSTEM INTEREST A

CURRENT ACTIVITY

SAMPLE ANALYSIS SEGMENT

USER DESCRIBES ENVIRONMENT AND EXPECTED MINERALS

USER VIEWS CROSS-SECTIONS OF THE DATA TO IDENTIFY REGIONS OF INTEREST

SYSTEM AUTOMATICALLY DETERMINES PARAMETERS FOR CLUSTERING OPERATION

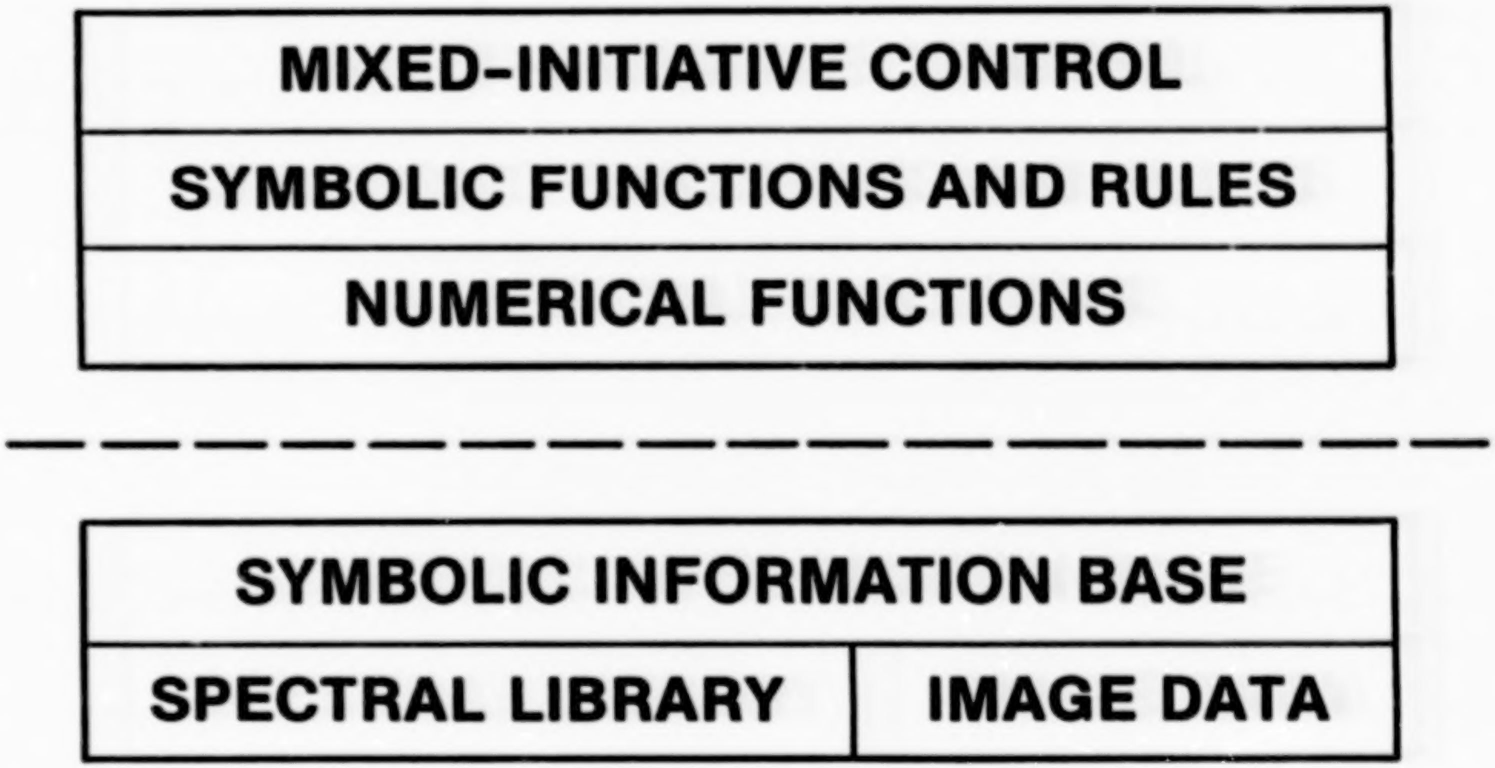
SYSTEM PERFORMS INITIAL CLUSTERING OPERATION

USER ALTERS CLUSTERING PARAMETERS AND DIRECTS SYSTEM TO RECLUSTER

•••

JPL

ORGANIZATION OF THE SPECTRUM SYSTEM



MIXED-INITIATIVE CONTROL

SYMBOLIC FUNCTIONS AND RULES

NUMERICAL FUNCTIONS

SYMBOLIC INFORMATION BASE

SPECTRAL LIBRARY

IMAGE DATA

FAULT-TOLERANT SOFTWARE: MODELING AND VALIDATION

Dave E. Eckhardt, Jr.
NASA Langley Research Center

NASA
Computer Science / Data Systems
Technical Symposium

November 18-20, 1986

BACKGROUND

- Advocacy/Research – 1975
  - Redundancy as a defense against residual faults in critical software
- Application
  - Airbus Industries A310 slat and flap control
  - Swedish state railroads traffic control system
  - Boeing 737–300 critical flight control functions
  - Boeing 757 yaw damper and stabilizer trim systems
  - Atomic Energy of Canada nuclear reactor shutdown system
  - NASA Space Shuttle mission critical functions
- Issues
  - Costs, overhead/performance, adjudication, applicability, independence
  - Fundamental issue, coincident failures

    Is software redundancy an effective strategy for achieving high reliability?

RESEARCH OBJECTIVES

- **Modeling**

  Develop model for analyzing redundant software where components are subject to coincident failure.

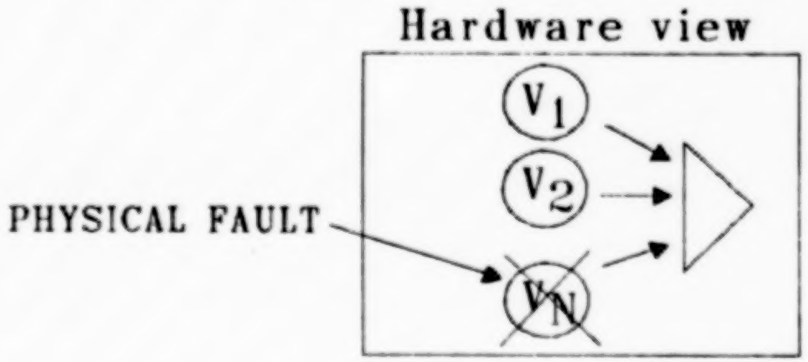
- **Experiments**

  Provide empirical data on the effectiveness of software redundancy (reliability gains).

- **Application**

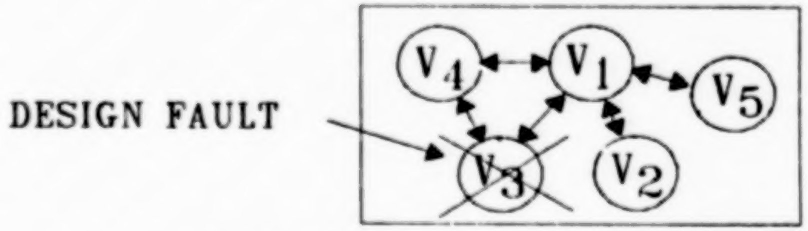
  Determine the applicability of systems theory to fault-tolerant software techniques.

MODELING REDUNDANCY


Hardware view
PHYSICAL FAULT
V1
V2
VN

$P_N = f(p)$

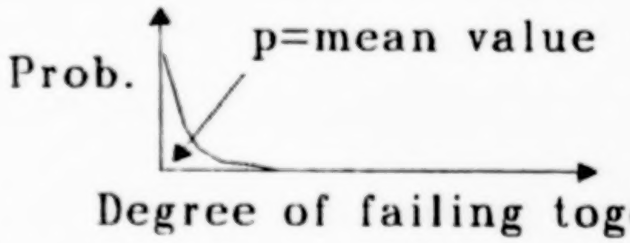
Are there complex dependencies among software versions?


DESIGN FAULT
V4
V1
V5
V3
V2

$P_N = f(p, \text{"correlation"})$

↑ meaning?

A distribution of failure intensity has meaning



$P_N = f(\text{Intensity Distr})$

COINCIDENT ERROR MODEL: A THEORETICAL BASIS FOR ANALYZING REDUNDANT SOFTWARE

Under the conditions that:

(1) components are selected from a random sample

(2) inputs are from a common usage distribtion Q

the expected probability of system failure P_N is:

$$P_N = \int \sum_{l=\frac{(N+1)}{2}}^{N} \binom{N}{l} y^l \, (1 - y)^{N-l} \, dG(y)$$

$$G(y) = \int_{\{x:\, \theta(x) \leq y\}} dQ \qquad \text{Intensity Distribution}$$

$\theta(x)$ = Intensity Function

COINCIDENT ERROR MODEL PROVIDES ANALYSIS OF:

- When redundancy is effective
- Limitations of redundancy
- Optimum level of redundancy
- Meaning of independence
- Impact of unclear, misleading specifications

What is the impact of a misleading specification on redundant software?

PAST RESPONSE:

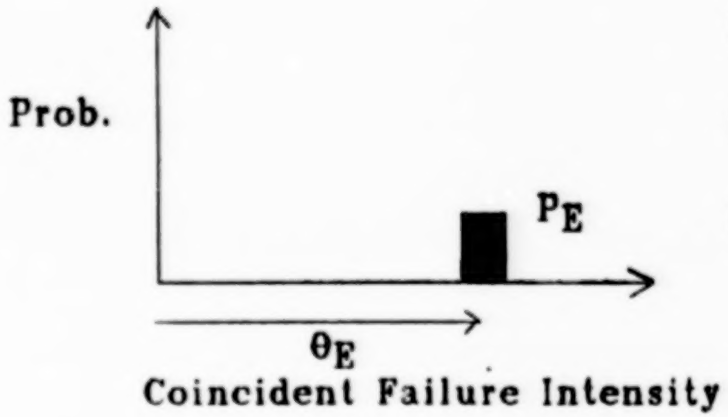
(after much agonizing about how to address problem)
"It makes the software worse"

CURRENT RESPONSE:

The concept of an intensity distribution provides a way to think about the problem.

What is the intensity coefficient?
(propensity to misinterpret)

What is the probability that an event will invoke the error?



E = Event which induces error

θ_E = Pr[incorrect interpretation]

P_E = Pr[E]

Suppose $g(.6) = 10^{-4}$ and $g(0) = 1 - 10^{-4}$

	P_N	
Number Versions	CEM Model	Independence Model
1	6.0×10^{-5}	6.0×10^{-5}
3	6.5×10^{-5}	1.1×10^{-8}
5	6.8×10^{-5}	2.2×10^{-14}

What causes redundancy to be an incorrect strategy?

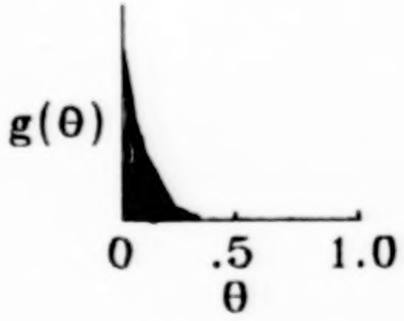
When is redundancy a more effective strategy than relying on a single version?

PAST RESPONSE: (Independence model)

$p < .5 \Longleftrightarrow p_N < p$ (and surely $p < .5$)

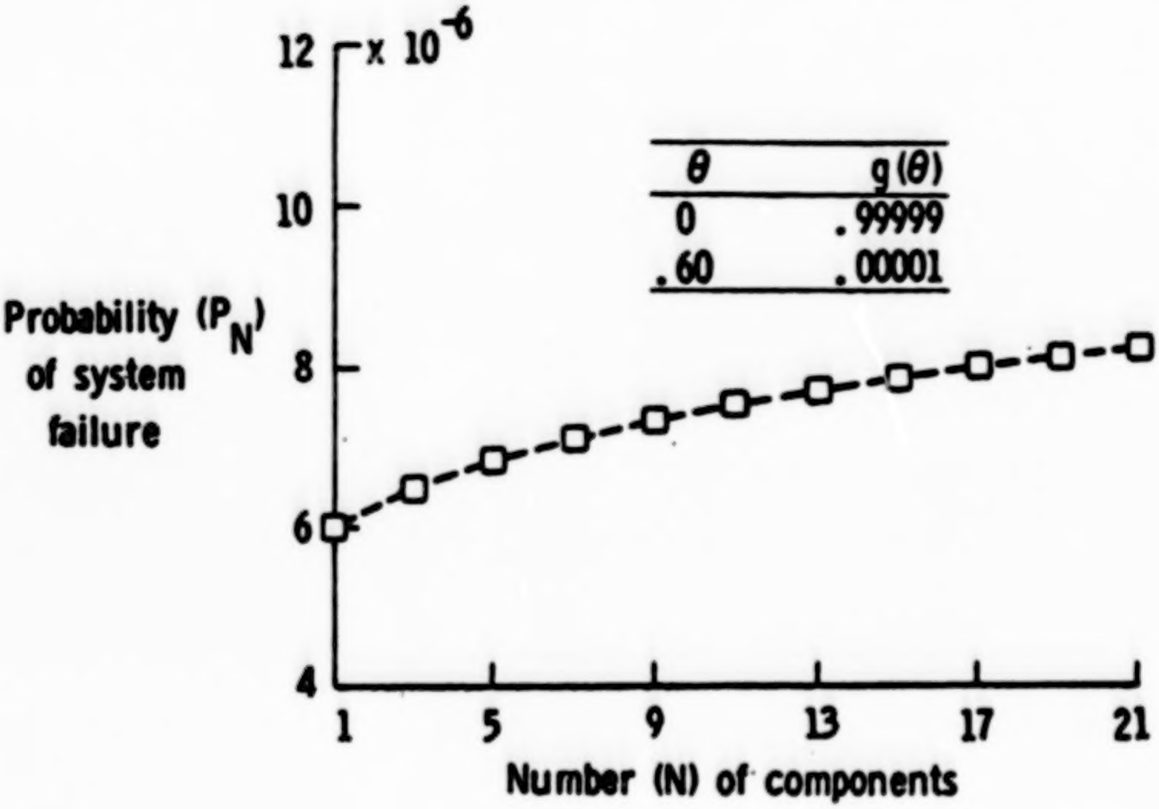
CURRENT RESPONSE: (Coincident Error Model)

If, for all feasible input conditions, the theoretical proportion of incorrect versions is less than .5 (i.e. a minority fail), this is sufficient (but not necessary) for redundancy to improve reliability.



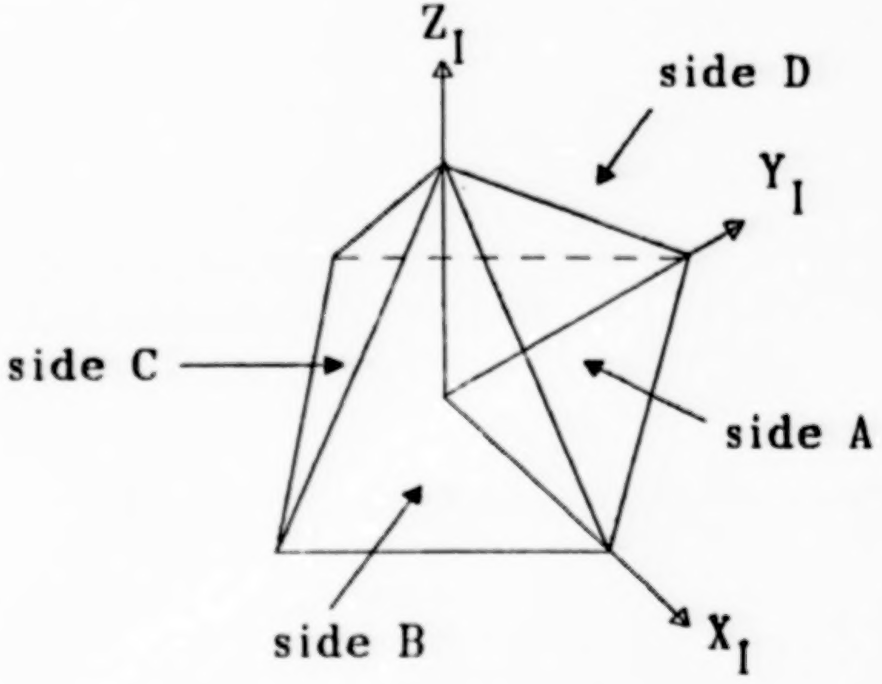
i.e intensity mass function bounded by .5

Further, $p < .5 \not\Rightarrow p_N < p$



FDI Logic Portion of the Redundancy Management & Vehicle State Estimation Software for a Skewed Sensor Array

Given: 6 edge relationships (AB, AC, AD, BC, BD, CD) on 4 faces (A, B, C, D). Determine faulty sensors.


Z_I
side D
Y_I
side C
side A
side B
X_I

2 orthogonal sensors/side

FDI Specification

Section 2. Functional Requirements

... "You are to assume that at most a single additional sensor will fail for a given execution. Therefore the face containing the failed sensor can be identified by comparing the faces pairwise and eliminating the face common to all out of tolerant condition."...

Section 8. Edge Vector Test

"Sensor failure detection and isolation is achieved by evaluating the status of the edge vector comparison tests:

$$\begin{tikzcd}[row sep=tiny]
|\varepsilon|_{AB} < \delta_{AB} & \\
|\varepsilon|_{AC} < \delta_{AC} & \\
|\varepsilon|_{AD} < \delta_{AD} & \text{violated edge conditions} \arrow[uul] \arrow[dl] \arrow[ddl] \\
|\varepsilon|_{BC} < \delta_{BC} & \\
|\varepsilon|_{BD} < \delta_{BD} & \\
|\varepsilon|_{CD} < \delta_{CD} &
\end{tikzcd}$$

For instance, if the second, third, and sixth relations are satisfied while the first, fourth, and fifth relations are violated, then either the x-accelerometer or y-accelerometer on face B must have failed."

FDI Interpretation

All sensors initially working

Group	Number of edge violations for face failure	Group size
1	1, 2, or 3	2
2	2 or 3	2
3	3	7
4	2 or 3 (1 causes all sensors to fail)	5
5	3 (1 or 2 causes all sensors to fail)	1
6	could not determine	3
	Total	20

THE IMPLEMENTATION AND USE OF ADA FOR FAULT-TOLERANT DISTRIBUTED SYSTEMS

NASA GRANT #NAG-1-260

WITH UNIVERSITY OF VIRGINIA

PRINCIPAL INVESTIGATOR:

DR. JOHN C. KNIGHT

PRESENTED BY: E. H. SENN
DATE: NOVEMBER 18-20, 1986

- o **INITIAL REQUIREMENT AND ASSUMPTIONS**

- REQUIREMENT: TO PROGRAM A DISTRIBUTED COMPUTER SYSTEM USING ADA, AND TO ACHIEVE SOFTWARE RECOVERY AND CONTINUATION OF SERVICE AFTER FAILURE OF NODE PROCESSORS IN THE NETWORK

- ASSUMPTIONS:
  - o A GENERAL DISTRIBUTED SYSTEM ARCHITECTURE
  - o A REPLICATED NETWORK COMMUNICATIONS STRUCTURE
  - o EXISTING INTRA-PROCESSOR ERROR DETECTION
  - o EXCESS PROCESSOR CAPACITY TO ALLOW SAME SERVICE RECOVERY FOLLOWING PROCESSOR FAILURES
  - o A COMMUNICATIONS PROTOCOL WHICH CONFORMS TO ISO STANDARD SEVEN-LAYER MODEL
  - o NO SYNTACTICAL CHANGES TO ADA

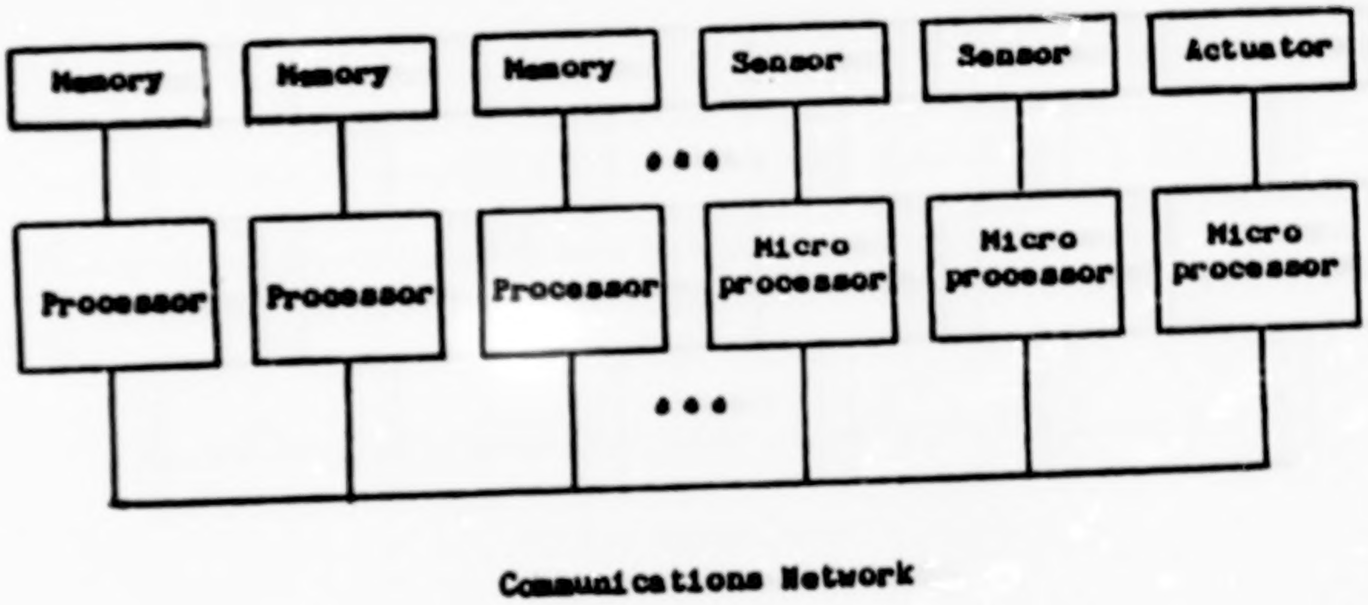

Memory
Memory
Memory
Sensor
Sensor
Actuator
Processor
Processor
Processor
Micro processor
Micro processor
Micro processor
Communications Network

Figure 1 - Distributed Architecture

- Current Working Methods and Approaches
  - Assumes a single Ada program consisting of multiple tasks distributed across available processors
  - Requires user-controlled task distribution method
  - Uses task abort semantics and exception interrupts to signal processor failure to system
  - Transparent versus controlled task recovery
    - Transparent/would always provide same service recovery
    - Controlled/could provide alternate or reduced service
  - Passive versus active failure detection methods
    - Passive/dependent on time-outs and assumptions
    - Active/uses periodic "heartbeat" software messages and generated task abort (failed) messages to signal processor failure and initiate recovery

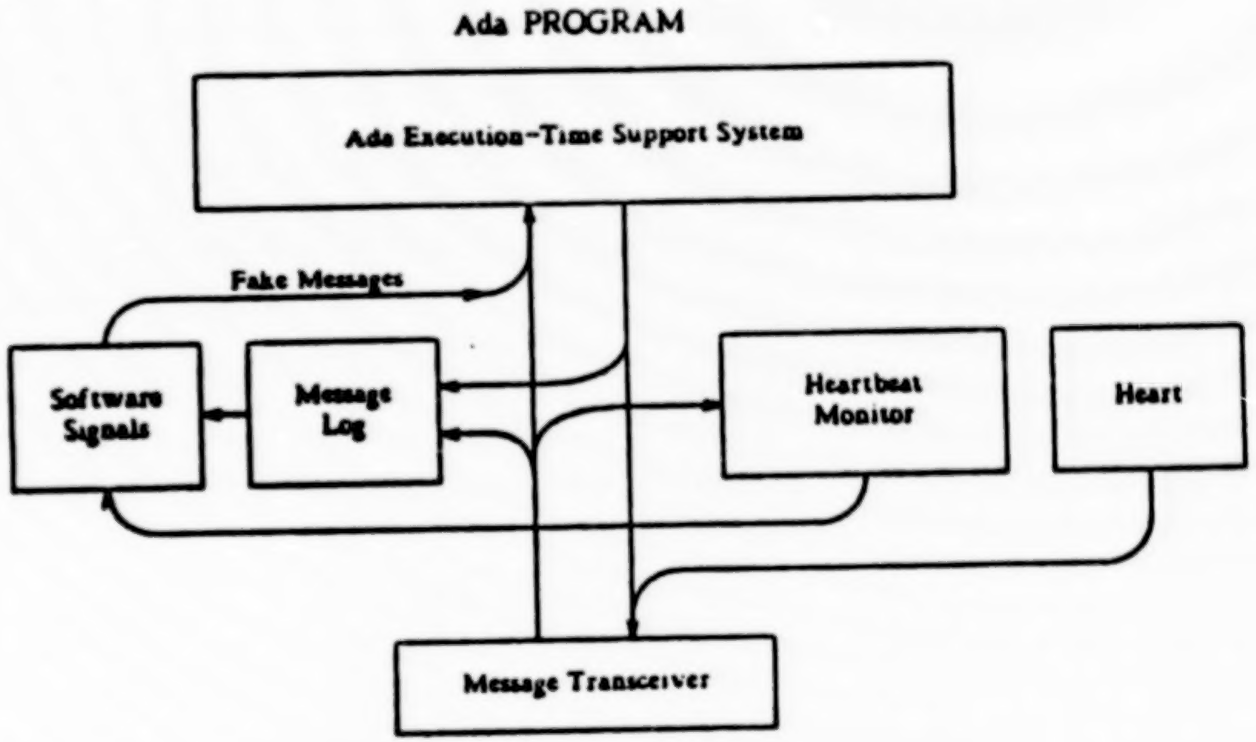

Ada PROGRAM
Ada Execution-Time Support System
Fake Messages
Software Signals
Message Log
Heartbeat Monitor
Heart
Message Transceiver

Fig. 2. Implementation Model For Each Processor

- o Current Testbed Status and Ada Problems
  - Testbed Status
    - o A usable FT/Ada testbed exists on a U. Va. VAX 11/750
    - o Uses enhanced run-time support, heartbeats, task aborts, and generated "fake" messages to signal processor failure and initiate recovery
    - o Ported to a network of Apollo micros (unsatisfactory)
    - o Plans to port to another micro network (suns?)
  - Ada Problems
    - o No provision for controlled distribution of tasks
    - o No syntax or semantics for handling processor failure
    - o No provisions for recovery/continuation actions
      - Damage assessment activities
      - Valid data (re) distribution/recovery
      - Task reconfiguration and restart

o Conclusions:

- We believe the known Ada problems can be, and are, resolved by the proposed mechanisms for hardware failure recovery

- We are planning to use a skeleton ATOPS Flight Code application to prove the Ada/testbed works as expected

- Ada makes no provisions for tolerance to software failures

- John Knight is proposing some extensions to Ada to support backward-error-recovery for software applications failures

  o Conversations/for communication and synchronization
  o A dialog/for controlled process communications and rollback
  o A colloquy/a set of dialogs and global acceptance tests

ARCHITECTURE RESEARCH AT ARC

DATA FLOW ARCHITECTURES

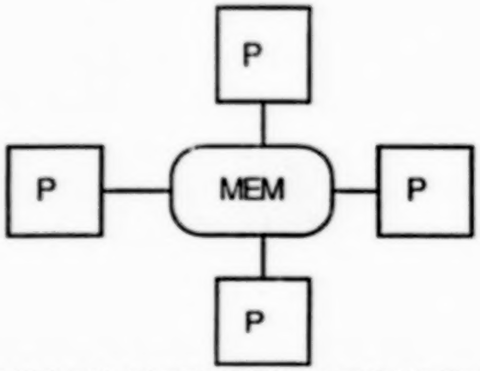
HOMOGENOUS MULTIPROCESSORS

HETEROGENEOUS MULTIPROCESSORS

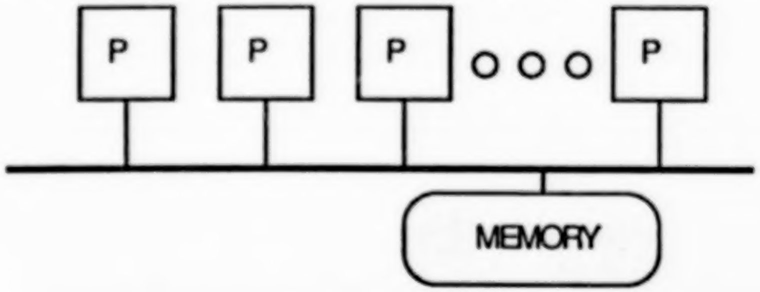
PERFORMANCE PREDICTION TOOLS

PROGRAMMING LANGAGES FOR MULTIPROCESSORS

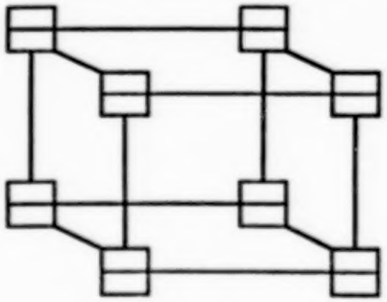
COMPUTER ARCHITECTURES



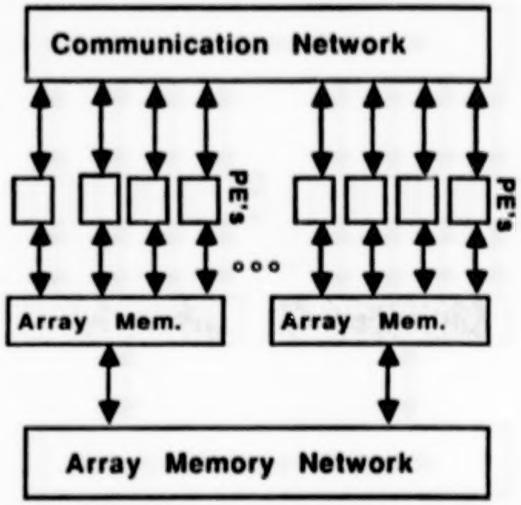
COMMON MEMORY ARCHITECTURE


P
P
P
O O O
P
MEMORY

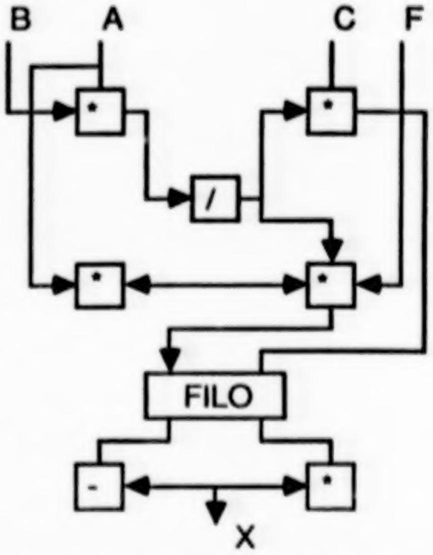
COMMON MEMORY BUS ARCHITECTURE



LOCAL MEMORY HYPERCUBE


Communication Network
PE's
PE's
o o o
Array Mem.
Array Mem.
Array Memory Network

DATA FLOW ARCHITECTURE



SYSTOLIC ARRAY ARCHITECTURE

Code

J = X + 1
K = Y + 2
Z = J + K

Sequential

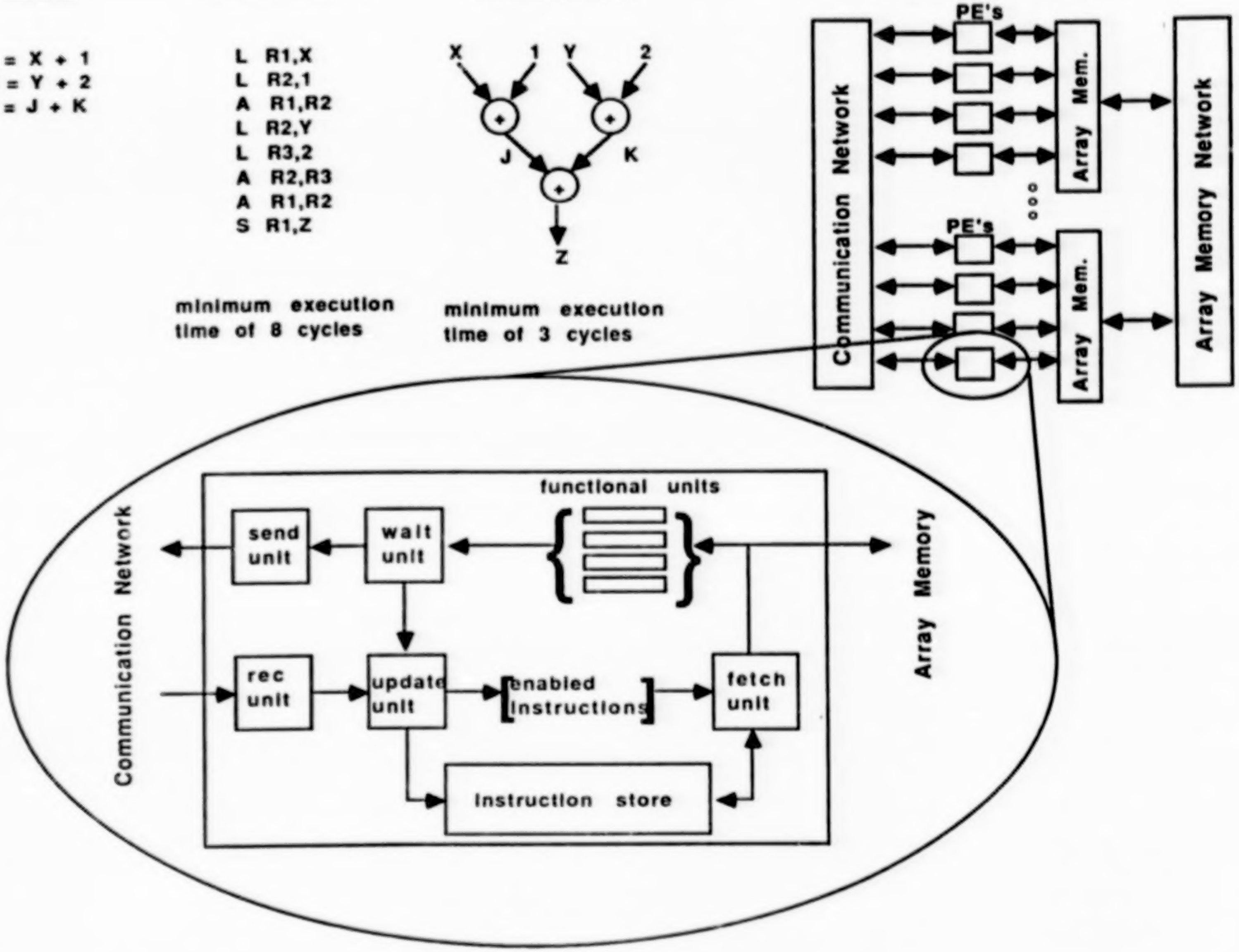
```
L  R1,X
L  R2,1
A  R1,R2
L  R2,Y
L  R3,2
A  R2,R3
A  R1,R2
S  R1,Z
```

minimum execution time of 8 cycles

Data Flow

minimum execution time of 3 cycles

Architecture


PE's
PE's
Communication Network
Array Mem.
Array Mem.
Array Memory Network
Communication Network
Array Memory
send unit
wait unit
functional units
rec unit
update unit
enabled instructions
fetch unit
instruction store

DATA FLOW

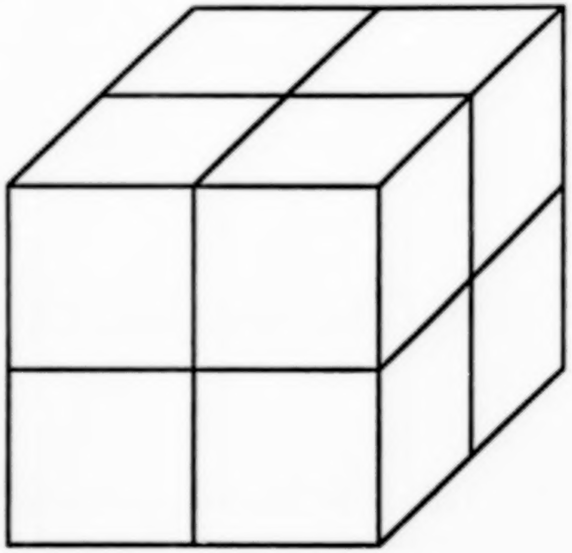
COMPLETED SIMULATOR

DEVELOPED INSTRUCTION PARTIONER WHICH IS ARCHITECTURE SENSITIVE

UPDATE UNIT IS THE BOTTLENECK

CURRENTLY SIMULATING THE NAS KERNALS

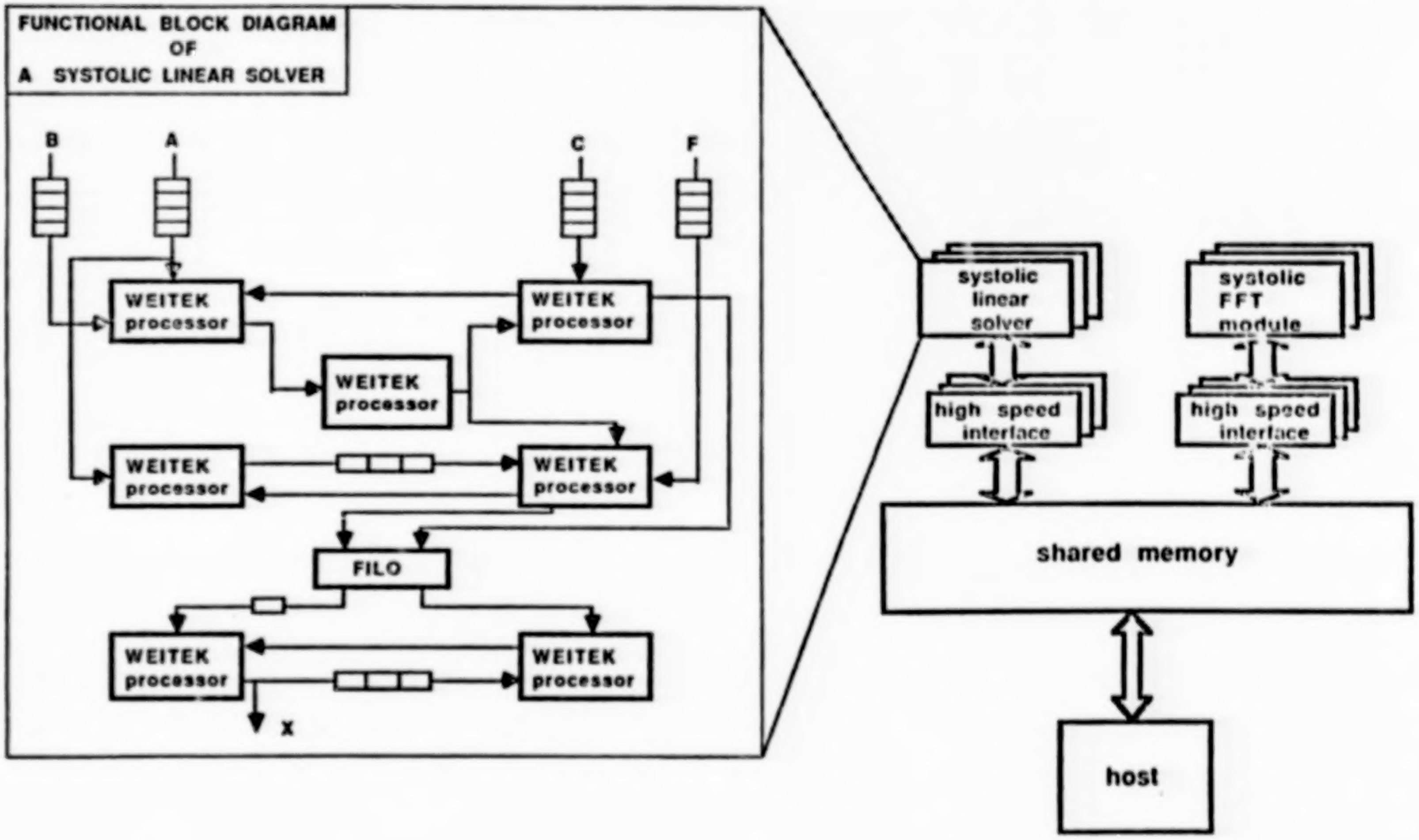
Mapping CFD Programs onto homogenous Multiprocessors



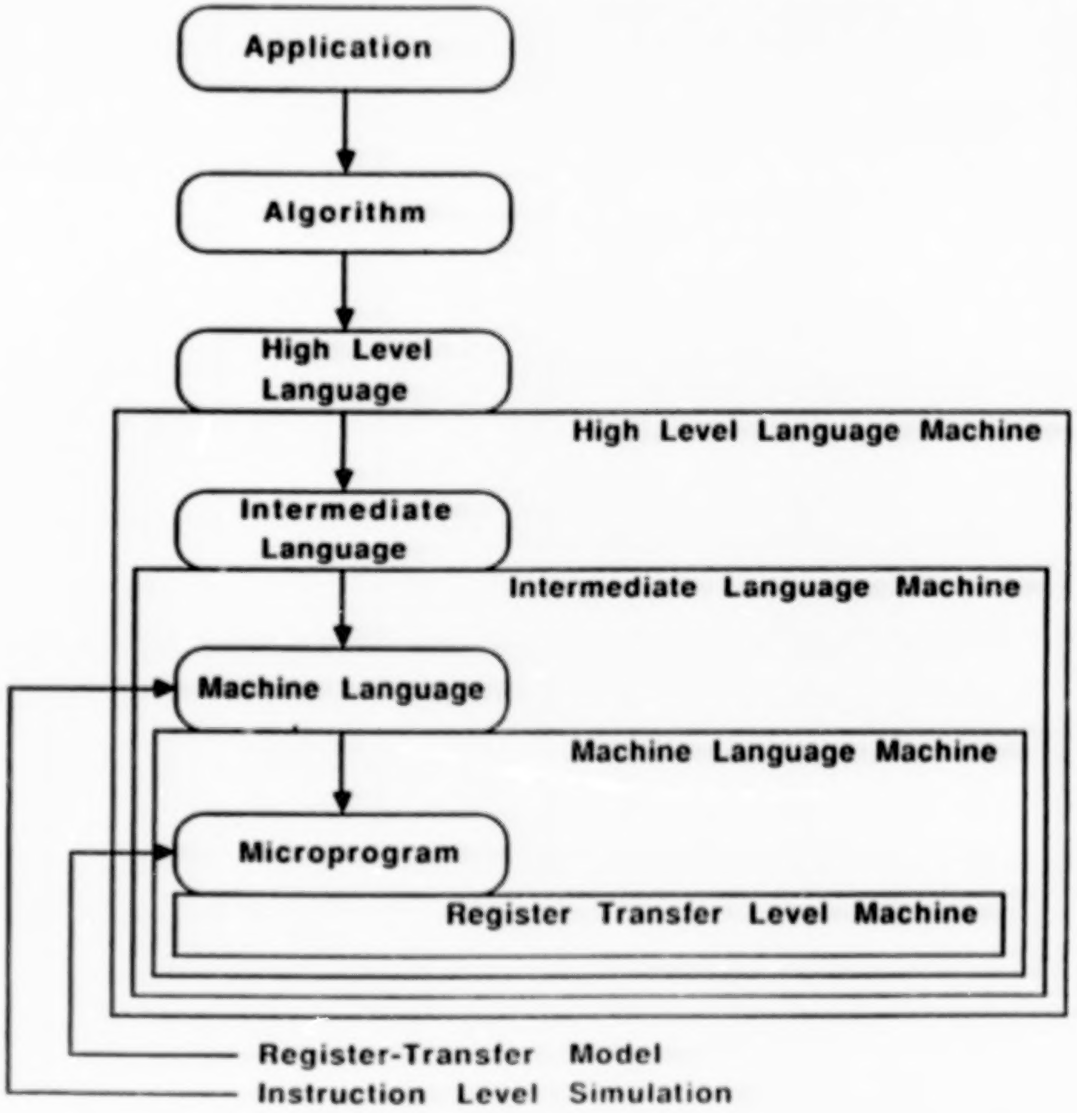
Spatially split algorithms (e.g. TWING, ARC3D, and LES) have been mapped onto common memory multiprocessors like the CRAY X-MP and CRAY 2.

Have mapped FLO52 onto the iPSC

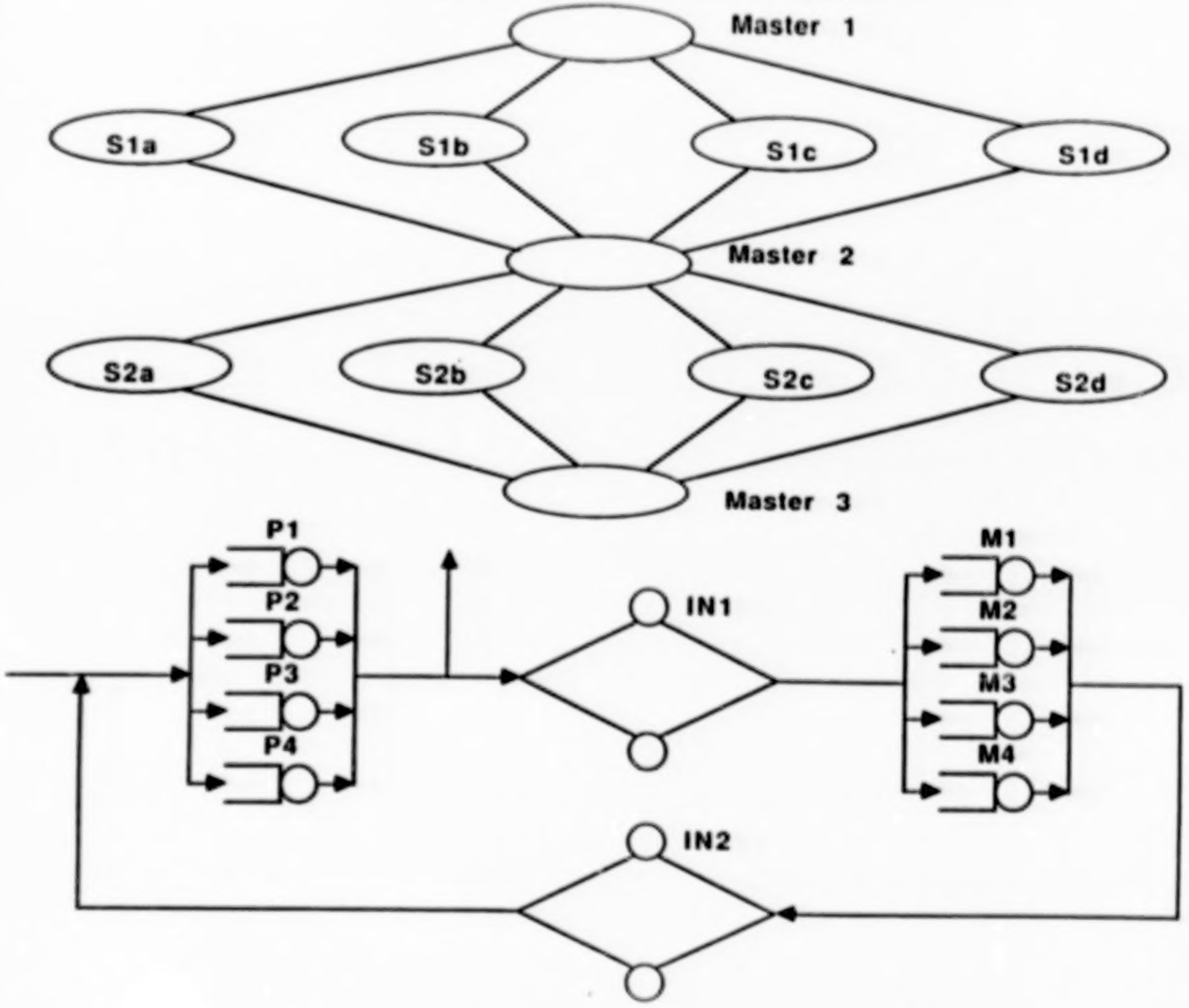
A LOW COST, HIGH SPEED
SYSTOLIC NAVIER-STOKES ATTACHED PROCESSOR (SNAP)


FUNCTIONAL BLOCK DIAGRAM
OF
A SYSTOLIC LINEAR SOLVER
B
A
C
F
WEITEK processor
WEITEK processor
WEITEK processor
WEITEK processor
WEITEK processor
FILO
WEITEK processor
WEITEK processor
X
systolic linear solver
systolic FFT module
high speed interface
high speed interface
shared memory
host

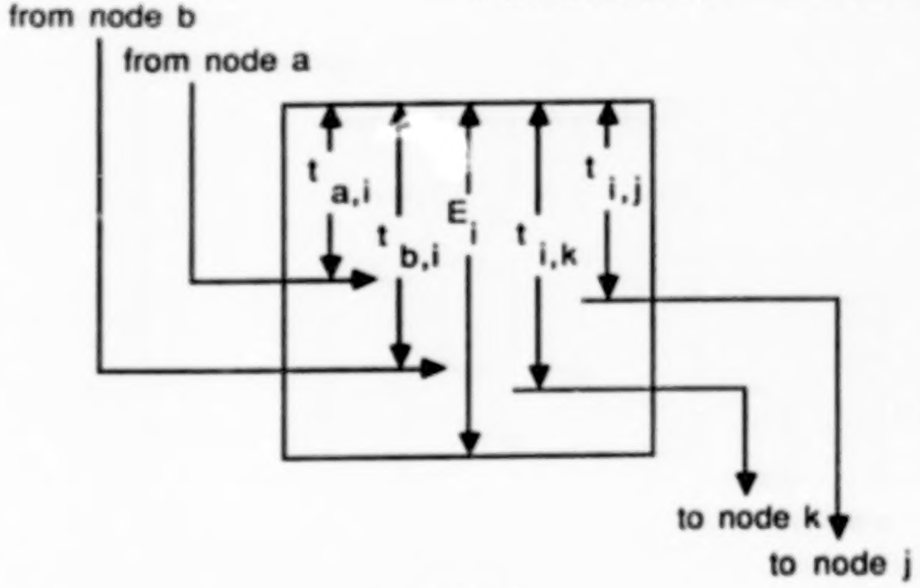
Level of Performance Modeling


Application
Algorithm
High Level Language
Intermediate Language
Machine Language
Microprogram
High Level Language Machine
Intermediate Language Machine
Machine Language Machine
Register Transfer Level Machine
Register-Transfer Model
Instruction Level Simulation

Queueing Network Models

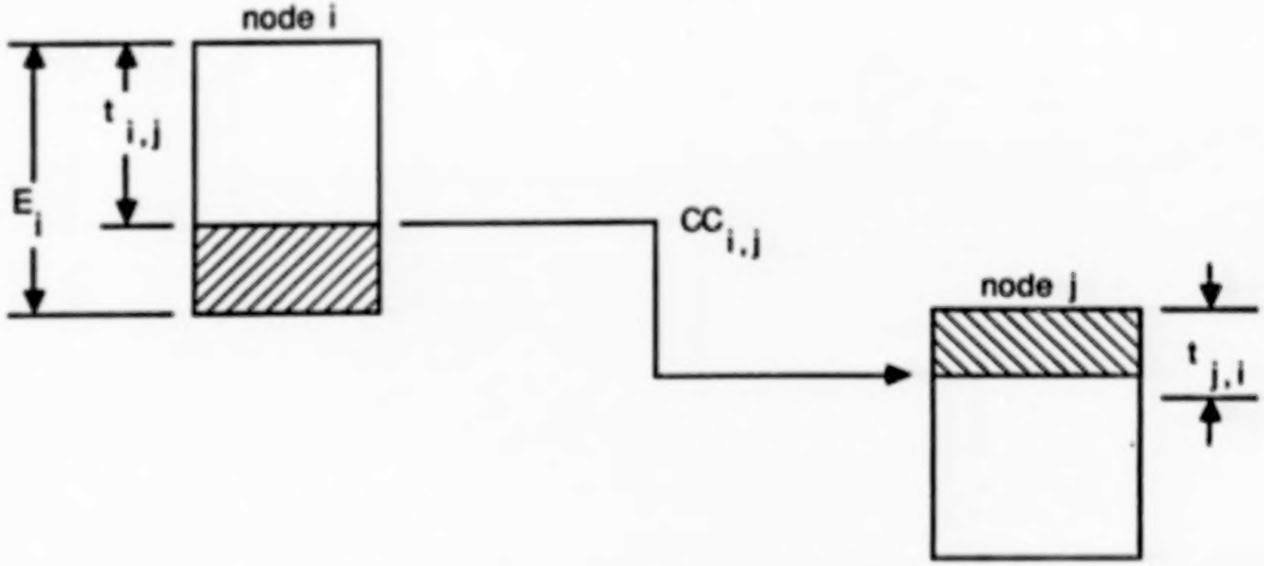

Master 1
S1a
S1b
S1c
S1d
Master 2
S2a
S2b
S2c
S2d
Master 3
P1
P2
P3
P4
IN1
M1
M2
M3
M4
IN2

Parallelism Detection


from node b
from node a
$t_{a,i}$
$t_{b,i}$
E_i
$t_{i,k}$
$t_{i,j}$
to node k
to node j

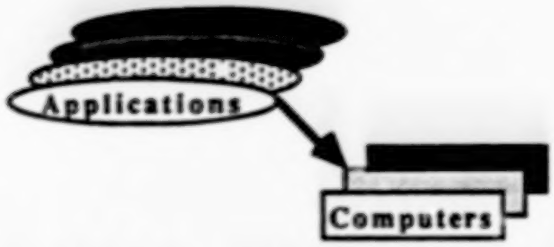
$t_{a,i}, t_{b,i}$ is the greatest possible computation time in node i where node i can be executed without the data package from node a and b.

$t_{i,j}, t_{j,k}$ is the earliest time when node i is ready to send the data package to node j and k.

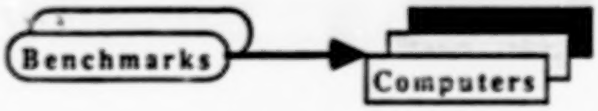

node i
E_i
$t_{i,j}$
$CC_{i,j}$
node j
$t_{j,i}$

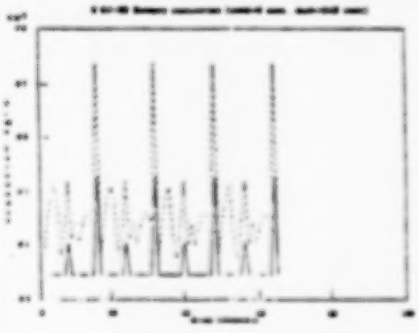
Benefit = $[(E_i - t_{i,j}) + t_{j,i}] - CC_{i,j}$ = saving from parallelism - communication overhead

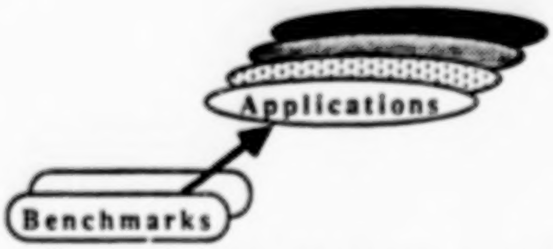
Performance Measurement Research


Applications
Computers

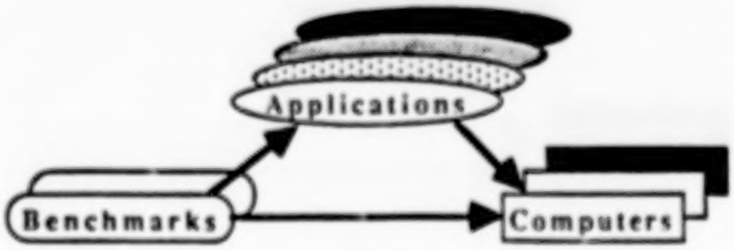
Livermore & NAS loops, Whetstones, Dhrystones


Benchmarks
Computers





Op counts, execution profiles, . . .


Applications
Benchmarks
Computers

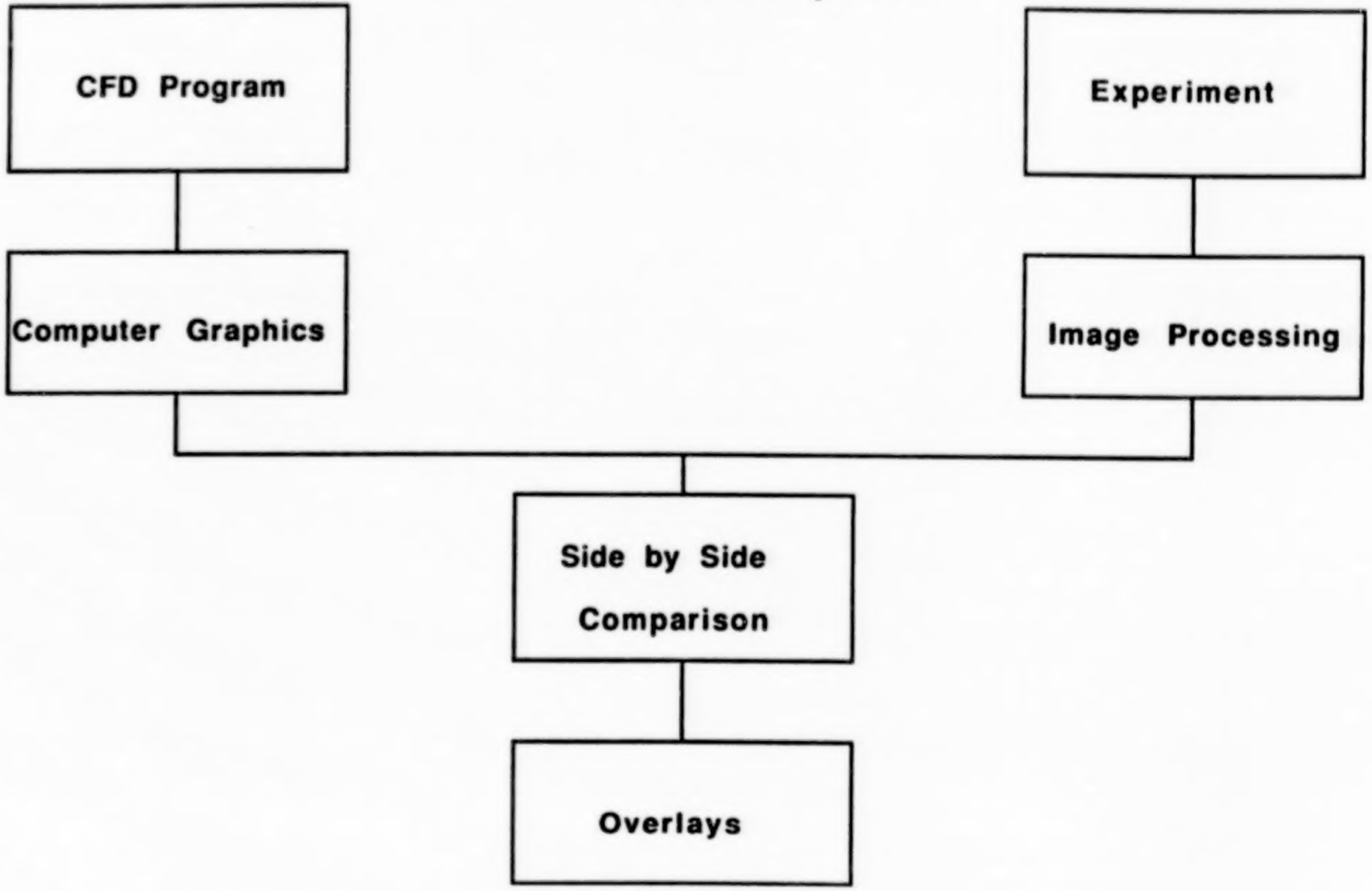
PROGRAMMING LANGUAGES FOR MULTIPROCESSORS

- MULTITASKING SUBROUTINES
    - SUBROUTINES TO CONTROL TASKS, FLAGS, AND LOCKS
    - PROVIDED BY CRAY RESEARCH BUT UNNATURAL FOR USERS
- MICROTASKING
    - DIRECTIVES TO A PRECOMPILER FOR LOOPS AND SEQUENTIAL CODE
    - PROVIDED BY CRAY RESEARCH BUT UNDEBUGGED AND FINE GRAIN
- COFORTRAN
    - COROUTINES WITH BARRIER SYNCRONIZATION
    - DEVELOPED AT ARC AND UNDER TEST BY ARC USERS

BENCHMARK	S3640 IFU+FPA	TI Explorer	SUN-3	LMI 2x2	MicroVax II
01. BOYER	16.81	27.00	13.38	xxx.xx	xxx.xx
02. BROWSE	39.90	48.06	57.28		
03. CTAK	7.64	3.73	2.08		
04. DDERIV	6.74	11.82	5.60		
05. DERIV	6.26	11.82	4.26		
06. DESTRUCTIVE	3.17	3.63	2.56		
07. DIV2 ITER	3.37	4.23	1.68		
DIV2 RECUR	5.53	7.20	3.18		
08. FFT	3.57	21.79	77.66		
09. FPRINT	7.79	7.49	1.66		
10. FREAD	11.23	10.12	3.74		
11. FRPOLY r 10	0.46	0.91	0.62		
r 15	3.05	5.98	3.92		
r2 10	2.61	1.52	4.16		
r2 15	20.47	11.53	46.06		
r3 10	0.46	1.16	1.56		
r3 15	2.31	6.98	10.82		
12. PUZZLE	15.50	26.38	8.08		
13. STAK	2.57	5.05	2.72		
14. TAK	0.60	1.68	0.70		
15. TAKL	6.43	16.23	2.52		
16. TAKR	0.60	1.69	0.98		
17. TPRINT	11.35	8.52	1.96		
18. TRAVERSE-INIT	11.23	20.38	22.10		
TRAVERSE-RUN	50.12	129.59	82.82		
19. TRIANGLE	247.47	425.42	138.80		

U.S. GOVERNMENT PRINTING OFFICE:1987-726-123/45001

Comparison of Computational and Experimental Results via Comuter Graphics



CFD Program

Computer Graphics

Experiment

Image Processing

Side by Side Comparison

Overlays

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA CP-2459	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Computer Sciences and Data Systems - Volume 1	5. Report Date March 1987
	6. Performing Organization Code RC
7. Author(s)	8. Performing Organization Report No.
	10. Work Unit No.
9. Performing Organization Name and Address Information Sciences and Human Factors Division NASA Office of Aeronautics and Space Technology	11. Contract or Grant No.
	13. Type of Report and Period Covered Conference Publication
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Presentation of OAST-supported work in progress were made by NASA personnel from Centers, Institutes, and Universities. The Computer Sciences subject material was grouped into the following categories: Software Engineering, University Grants, Institutes and Applications. The material presented under Data Systems was not categorized, as such. The Symposium was held at the National Conference Center in Williamsburg, Virginia from November 18 to 20, 1986. The Symposium schedule and the list of attendees are included.

17. Key Words (Suggested by Author(s)) Computer Sciences Data Systems Software Engineering University Grants Applications Expert Systems Institutes	18. Distribution Statement Unclassified-Unlimited Subject Category 61

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 350	22. Price A15

NASA FORM 1626 OCT 86

END

DATE

FILMED

MAY 29 1987